This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

ISBNet: a 3D Point Cloud Instance Segmentation Network with Instance-aware Sampling and Box-aware Dynamic Convolution

Tuan Duc Ngo   Binh-Son Hua   Khoi Nguyen
VinAI Research, Hanoi, Vietnam
{v.tuannd42, v.sonhb, v.khoindm}@vinai.io

Abstract

Existing 3D instance segmentation methods are predominated by the bottom-up design – manually fine-tuned algorithm to group points into clusters followed by a refinement network. However, by relying on the quality of the clusters, these methods generate susceptible results when (1) nearby objects with the same semantic class are packed together, or (2) large objects with loosely connected regions. To address these limitations, we introduce ISBNet, a novel cluster-free method that represents instances as kernels and decodes instance masks via dynamic convolution. To efficiently generate high-recall and discriminative kernels, we propose a simple strategy named Instance-aware Farthest Point Sampling to sample candidates and leverage the local aggregation layer inspired by PointNet++ to encode candidate features. Moreover, we show that predicting and leveraging the 3D axis-aligned bounding boxes in the dynamic convolution further boosts performance. Our method set new state-of-the-art results on ScanNetV2 (55.9), S3DIS (60.8), and STPLS3D (49.2) in terms of AP and retains fast inference time (237ms per scene on ScanNetV2). The source code and trained models are available at https://github.com/VinAIResearch/ISBNet.

1. Introduction

3D instance segmentation (3DIS) is a core problem of deep learning in the 3D domain. Given a 3D scene represented by a point cloud, we seek to assign each point with a semantic class and a unique instance label. 3DIS is an important 3D perception task and has a wide range of applications in autonomous driving, augmented reality, and robot navigation where point cloud data can be leveraged to complement the information provided by 2D images. Compared to 2D image instance segmentation (2DIS), 3DIS is arguably harder due to much higher variations in appearance and spatial extent along with unequal distribution of point cloud, i.e., dense near object surface and sparse elsewhere. Thus, it is not trivial to apply 2DIS methods to 3DIS.

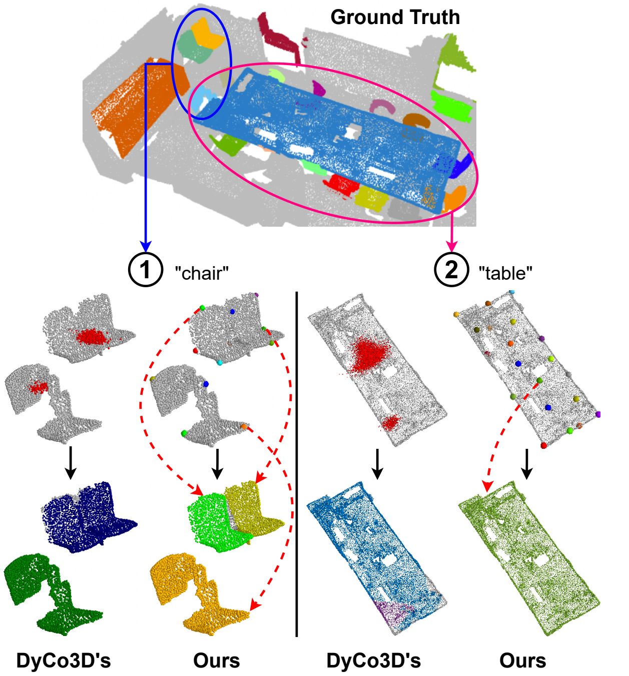

Figure 1. In DyCo3D [16], kernel prediction quality is greatly affected by the centroid-based clustering algorithm which has two issues: ① mis-grouping nearby instances and ② over-segment a large object into multiple fragments. Our method addresses these issues by *instance-aware point sampling*, achieving far better results. Each sample point aggregates information from its local context to generate a kernel for predicting its own object mask, and the final instances will be filtered and selected by an NMS.

A typical approach for 3DIS, DyCo3D [16], adopts dynamic convolution [33, 37] to predict instance masks. Specifically, points are clustered, voxelized, and passed through a 3D Unet to generate instance kernels for dynamic convolution with the feature of all points in the scene. This approach is illustrated in Fig. 2 (a). However, this approach has several limitations. First, the clustering algorithm heavily relies on the centroid-offset prediction whose quality deteriorates significantly when: (1) objects are densely packed so that two objects can be mistakenly grouped together as one object, or (2) large objects whose parts are loosely connected resulting in different objects when clustered. These two scenarios are visualized in Fig. 1. Second, the points' feature mostly encodes object appearance which is not distinct enough for separating different instances, especially between objects having the same semantic class.

To address the limitations of DyCo3D [16], we propose ISBNet, a cluster-free framework for 3DIS with **I**nstance-aware Farthest Point **S**ampling and **B**ox-aware Dynamic Convolution. First, we revisit the Farthest Point Sampling (FPS) [10] and the clustering method in [5, 16, 34] and find that these algorithms generate considerably low instance recall. As a result, many objects are omitted in the subsequent stage, leading to poor performance. Motivated by this, we propose our Instance-aware Farthest Point Sampling (IA-FPS), which aims to sample query candidates in a 3D scene with high instance recall. We then introduce our Point Aggregator, incorporating the IA-FPS with a local aggregation layer to encode semantic features, shapes, and sizes of instances into instance-wise features.

Additionally, the 3D bounding box of the object is an existing supervision but has not yet been explored in the 3D instance segmentation task. Therefore, we add an auxiliary branch to our model to jointly predict the axis-aligned bounding box and the binary mask of each instance. The ground-truth axis-aligned bounding box is deduced from the existing instance mask label. Unlike Mask-DINO [25] and CondInst [33], where the auxiliary bounding box prediction is just used as a regularization of the learning process, we leverage it as an extra geometric cue in the dynamic convolution, thus further boosting the performance of the instance segmentation task.

To evaluate the performance of our approach, we conduct extensive experiments on three challenging datasets: ScanNetV2 [8], S3DIS [1], and STPLS3D [4]. ISBNet not only achieves the highest accuracy among these three datasets, surpassing the strongest method by +2.7/3.4/3.0 on ScanNetV2, S3DIS, and STPLS3D, but also demonstrates to be highly efficient, running at 237ms per scene on ScanNetV2.

In summary, the contributions of our work are as follows:

- We propose ISBNet, a cluster-free paradigm for 3DIS, that leverages Instance-aware Farthest Point Sampling and Point Aggregator to generate an instance feature set.
- We first introduce using the axis-aligned bounding box as an auxiliary supervision and propose the Box-aware Dynamic Convolution to decode instance binary masks.
- ISBNet achieves state-of-the-art performance on three different datasets: ScanNetV2, S3DIS, and STPLS3D without comprehensive modifications of the model architecture and hyper-parameter tuning for each dataset.

In the following, Sec. 2 reviews prior work; Sec. 3 specifies our approach; and Sec. 4 presents our implementation details and experimental results. Sec. 5 concludes with some remarks and discussions.

2. Related Work

2D image instance segmentation (2DIS) concerns assigning each pixel in the image with one of the instance labels and semantic labels. Its approaches can be divided into three groups: proposal-based, proposal-free, and DETR-based approaches. For proposal-based methods [2, 15, 22], an object detector, e.g., Faster-RCNN [30] is leveraged to predict object bounding boxes to segment the foreground region inside detected boxes. For proposal-free methods [33, 36, 37], SOLO [36, 37] and CondInst [33] predict the instance kernels for the dynamic convolution with the feature maps to generate instance masks. For DETR-based methods [6, 7, 13, 25], Mask2Former [7] and Mask-DINO [25] employ the transformer architecture with instance queries to obtain the segmentation for each instance. Compared to 3DIS, 2DIS is arguably easier due to the structured, grid-based, and dense properties of 2D images. Hence, it is not trivial to adapt a 2DIS method to 3DIS.

3D point cloud instance segmentation (3DIS) methods are interested in labeling each point in a 3D point cloud with a semantic class and a unique instance ID. They can be categorized into proposal-based, clustering-based, and dynamic convolution-based methods. Cluster

Proposal-based methods [18, 39, 40] first detect 3D bounding boxes and then segment the foreground region inside each box to form instances. 3D-SIS [18] adapts the Mask R-CNN architecture to 3D instance segmentation and jointly learns features from two modalities of RGB images and 3D point clouds. 3D-BoNet [39] predicts a fixed number of 3D bounding boxes from a global feature vector summarizing the content of the scene and then segments the foreground points inside each box. A limitation of this approach is that the performance of instance masks heavily depends on the quality of 3D bounding boxes which is very unstable due to the huge variation and uneven distribution of 3D point cloud.

Clustering-based methods [5, 11, 16, 21, 28, 34, 35, 42] learn latent embeddings that facilitate grouping points into instances. PointGroup [21] predicts the 3D offset from each point to its instance's centroid and obtains the clusters from two point clouds: original points and centroid-shifted points. HAIS [5] proposes a hierarchical clustering method where a small cluster can be filtered out or absorbed by a larger cluster. SoftGroup [34] proposes a soft-grouping strategy in which each point can belong to multiple clusters with different semantic classes to alleviate the semantic prediction error. One of the limitations of the clustering-based approach is that the quality of the instance masks significantly depends on the quality of the clustering, i.e., the centroid prediction, which is greatly unreliable especially when testing objects considerably differ from training objects in spatial extent.

Dynamic convolution-based methods [16, 17, 38] over-

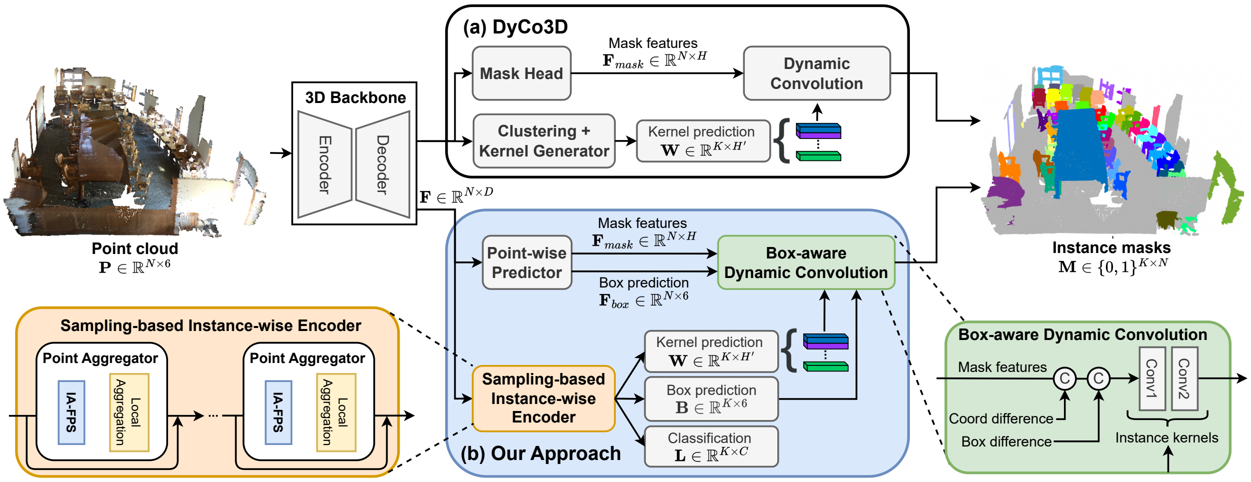
(a) DyCo3D
Mask Head
Mask features
$\mathbf{F}_{mask} \in \mathbb{R}^{N\times H}$
Dynamic Convolution
Clustering + Kernel Generator
Kernel prediction
$\mathbf{W} \in \mathbb{R}^{K\times H'}$
3D Backbone
Encoder
Decoder
$\mathbf{F} \in \mathbb{R}^{N\times D}$
Point cloud
$\mathbf{P} \in \mathbb{R}^{N\times 6}$
Instance masks
$\mathbf{M} \in \{0,1\}^{K\times N}$
Point-wise Predictor
Mask features
$\mathbf{F}_{mask} \in \mathbb{R}^{N\times H}$
Box prediction
$\mathbf{F}_{box} \in \mathbb{R}^{N\times 6}$
Box-aware Dynamic Convolution
Sampling-based Instance-wise Encoder
Kernel prediction
$\mathbf{W} \in \mathbb{R}^{K\times H'}$
Box prediction
$\mathbf{B} \in \mathbb{R}^{K\times 6}$
Classification
$\mathbf{L} \in \mathbb{R}^{K\times C}$
(b) Our Approach
Sampling-based Instance-wise Encoder
Point Aggregator
IA-FPS
Local Aggregation
...
Point Aggregator
IA-FPS
Local Aggregation
Box-aware Dynamic Convolution
Mask features
C
C
Conv1
Conv2
Coord difference
Box difference
Instance kernels

Figure 2. Overall architectures of DyCo3D [16] (block (a)) and our approach (block (b)) for 3DIS. Given a point cloud, a 3D backbone is employed to extract per-point features. For DyCo3D, it first groups points into clusters based on the predicted object centroid from each point to generate a kernel for each cluster. In the meantime, the mask head transforms the per-point features into mask features for dynamic convolution. For our approach ISBNet, we replace the clustering algorithm with a novel sampling-based instance-wise encoder to obtain faster and more robust kernel, box, and class predictions. Furthermore, a point-wise predictor replaces the mask head of DyCo3D to output the mask and box features for a new box-aware dynamic convolution to produce more accurate instance masks.

come the limitations of proposal-based and clustering-based methods by generating kernels and then using them to convolve with the point features to generate instance masks. DyCo3D [16] adopts the clustering algorithm in [21] to generate kernels for dynamic convolution. PointInst3D [17] uses farthest-point sampling to replace the clustering in [16] in order to generate kernels. DKNet [38] introduces candidate mining and candidate aggregation to generate more discriminative instance kernels for dynamic convolution.

Our approach is a dynamic convolution-based method with two important improvements in kernel generation and dynamic convolution. Particularly, in the former, we propose a new instance encoder combining instance-aware farthest-point sampling with a point aggregation layer to generate kernels to replace clustering in DyCo3D [16]. In the latter, instead of only using the appearance feature for dynamic convolution, we additionally enhance that feature with a geometry cue namely bounding box prediction.

3. Our Approach

Problem statement: Given a 3D point cloud $\mathbf{P} \in \mathbb{R}^{N\times 6}$ where N is the number of points, and each point is represented by a 3D position and RGB color vector. We aim to segment the point cloud into K instances that are represented by a set of binary masks $\mathbf{M} \in \{0,1\}^{K\times N}$ and a set of semantic labels $\mathbf{L} \in \mathbb{R}^{K\times C}$, where C is the number of semantic categories.

Our method consists of four main components: a 3D backbone, a point-wise predictor, a sampling-based instance-wise encoder, and a box-aware dynamic convolution. The 3D backbone takes a 3D point cloud as input to extract per-point features. Our backbone network extracts feature $f^{(i)} \in \mathbb{R}^D$ where $i = 1, \ldots, N$ for each point of the input point cloud. We follow previous methods [5, 34, 38] to adopt a U-Net with sparse convolutions [12] as our backbone. The point-wise predictor takes per-point features $\mathbf{F} \in \mathbb{R}^{N\times D}$ from the backbone and transforms them into point-wise semantic predictions, axis-aligned bounding box predictions $\mathbf{F}_{box} \in \mathbb{R}^{N\times 6}$, and mask features $\mathbf{F}_{mask} \in \mathbb{R}^{N\times H}$ for box-aware dynamic convolution. The sampling-based instance-wise encoder (Sec. 3.1) processes point-wise features to generate instance kernels, instance class labels, and bounding box parameters. Finally, the box-aware dynamic convolution (Sec. 3.2) gets the instance kernels and the mask features with the complementary box prediction to generate the final binary mask for each instance. An overview of our method is illustrated in Fig. 2.

# sampling points	2048	512	256	128
FPS	99.3%	93.3%	85.4%	71.3%
IA-FPS	100%	98.4%	94.5%	89.2%
Clustering	75.5%	75.5%	75.5%	75.5%

Table 1. The recall of different sampling methods on ScanNetV2 validation set. FPS is the standard Farthest Point Sampling, IA-FPS is our proposed Instance-aware Farthest Point Sampling. Clustering is the algorithm used in [5, 16, 34] and its recall does not depend on the number of sampling points.

3.1. Sampling-based Instance-wise Encoder

Given per-point features $\mathbf{F} \in \mathbb{R}^{N\times D}$ output from the backbone, we aim to produce instance-wise features $\mathbf{E} \in \mathbb{R}^{K\times D}$ where $K \ll N$. The instance-wise feature $\mathbf{E}$

is then used to predict the instance classification scores $\mathbf{L} \in \mathbb{R}^{K \times C}$, instance boxes $\mathbf{B} \in \mathbb{R}^{K \times 6}$, and instance kernels $\mathbf{W} \in \mathbb{R}^{K \times H'}$ where H' is decided by the sizes of the convolutional layers in dynamic convolution.

Typically, one can employ Farthest Point Sampling (FPS) [10] to sample a set of K candidates to generate instance kernels as in [17]. FPS greedily samples points in 3D coordinates by choosing the next points farthest away from the previous sampled ones using the pairwise distance. However, this sampling technique is inferior. First, there are many points belonging to the background categories among the K sampled candidates by FPS, wasting computational resources. Second, large objects dominate the number of sampled points hence no point is sampled from small objects. Third, the point-wise features cannot capture the local context to create instance kernels. We provide analysis in Tab. 1 to validate this observation. Particularly, we calculate the recall of the number of instances predicted by the kernels against the total ground truth instances. The recall value should be large as we expect good coverage of the clustered or sampled points on the ground-truth instances. However, as can be seen, previous methods have low recall which can be explained by that these methods do not consider instances for point clustering or sampling.

To address this issue, we propose a novel sampling-based instance-wise encoder that takes instances into account in the point sampling step. Inspired by the Set Abstraction in PointNet++ [29], we specify our instance encoder comprising a sequence of Point Aggregator (PA) blocks whose components are Instance-Aware FPS (IA-FPS) to sample candidate points covering as many foreground objects as possible and a local aggregation layer to capture the local context so as to enrich the candidate features individually. We visualize the PA in the orange block in Fig. 2 and detail our sampling below.

Instance-aware FPS. Our sampling strategy is to sample foreground points to maximally cover all instances regardless of their sizes. To achieve this goal, we opt for an iterative sampling technique as follows. Specifically, candidates are sampled from a set of points that are neither background nor chosen by previous sampled candidates. We use the point-wise semantic prediction to estimate the probability for each point to be background $m_{(0)}^{(i)} \in [0, 1]$. We also use the instance masks generated by previous k-th candidate $m_{(k)}^{(i)} \in [0, 1]$. The FPS is leveraged to sample points from the set of points $\mathbf{P}' \subset \mathbf{P}$:

$$\mathbf{P}' = \left\{ p^{(i)} \in \mathbf{P} \middle| \min_{k=0..K'} \left(1 - m_{(k)}^{(i)}\right) > \tau \right\}, \qquad (1)$$

where K' is the number of already chosen candidates and τ, is the hyper-parameter threshold.

Practically, in training, since the instance mask prediction is not good enough for guiding the instance sampling, K candidates are sampled altogether at once from the predicted foreground mask $1 - m_{(0)}^{(i)} > \tau$. On the other hand, in testing, we iteratively sample smaller chunks $\{\kappa_1, \dots, \kappa_T\}$ one by one such that the subsequent chunks will be sampled from neither the background points nor points belonging to predicted masks of previous chunks. By doing so, the recall rate of IA-FPS improves a lot as shown in Tab. 1.

Local Aggregation Layer. For each candidate k, the local aggregation layer encodes and transforms the local context into its instance-wise features. Specifically, Ball-query is employed [29] to collect its Q local neighbors as the local features $\mathbf{F}_{local}^{(k)} \in \mathbb{R}^{Q \times D}$. Also, the relative coordinates between the candidates k and their neighbors q are computed and normalized with the neighborhood radius r to form the local coordinates, or $\mathbf{P}_{local}^{(k)} \in [-1, 1]^{Q \times 3}$. Next, we use an MLP layer to transform the local features $\mathbf{F}_{local}^{(k)}$ and the local coordinates $\mathbf{P}_{local}^{(k)}$ into the instance-wise features $e^{(k)}$ of candidate k. We also add a residual connection with the original features $f^{(k)}$ to avoid gradient vanishing. Concretely, the instance-wise feature can be computed as:

$$e^{(k)} = f^{(k)} + \max_{q} \left(\text{MLP} \left(\left[\mathbf{F}_{local}^{(k)}; \mathbf{P}_{local}^{(k)} \right] \right) \right), \qquad (2)$$

where $[\cdot; \cdot]$ denotes the concatenation operations. From $\mathbf{E}$, a linear layer is used to predict instance classification scores $\mathbf{L}$, instance boxes $\mathbf{B}$, and instance kernels $\mathbf{W}$.

It is worth noting that, to obtain the instance-wise features $\mathbf{E}$, instead of using a single PA block, we propose a progressive way, by sequentially applying multiple PA blocks. In this way, the subsequent block will sample from the smaller number of points sampled by the previous block. Doing so has the same effect as stacking multiple convolutional layers in 2D images in order to increase the receptive field.

3.2. Box-aware Dynamic Convolution

In the dynamic convolution of [16, 17, 38], for each candidate k, the relative position of all points w.r.t k, $\mathbf{F}_{pos}^{(k)} \in \mathbb{R}^{N \times 3}$ and the point-wise mask features $\mathbf{F}_{mask} \in \mathbb{R}^{N \times H}$ are concatenated and convolved with instance kernels $w^{(k)}$ to obtain the instance binary mask $\widehat{m}^{(k)} = \text{Sigmoid}\left(\text{Conv}\left(\left[\mathbf{F}_{mask}; \mathbf{F}_{pos}^{(k)}\right]; w^{(k)}\right)\right)$, where Conv is implemented as several convolutional layers. However, we would argue that only using the mask features and positions is sub-optimal. For example, points near the boundary of adjacent objects whose class is the same are indistinguishable from each other when only using the mask features and positions in 3D.

On the other hand, 3D bounding box delineates the shape and size of an object, which provides an important geometric cue for the prediction of object masks in instance segmentation. Our method uses bounding box predictions as an auxiliary task that regularizes instance segmentation training. Particularly, for each point, we propose to regress the axis-aligned bounding box deduced from the object mask. The predicted boxes $\mathbf{F}_{box} \in \mathbb{R}^{N\times 6}$ are then used to condition the mask feature to generate kernels for the box-aware dynamic convolution (see the green block in Fig. 2). Each bounding box is parameterized by a 6D vector $f^{(i)}_{box} = (x_1, y_1, z_1, x_2, y_2, z_2)$ that represents the minimum and maximum bound of the point coordinates of an instance. It is worth noting that we choose to use axis-aligned bounding boxes because the ground-truth boxes are basically available for free as they can be easily constructed from ground-truth instance annotations.

Therefore, we propose to use the predicted boxes as an additional geometric cue in dynamic convolution, giving the name of our proposed box-aware dynamic convolution. Intuitively, two points will belong to the same object if their predicted boxes are similar. Our final instance mask $\widehat{m}^{(k)} \in [0,1]^{1\times N}$ of the k-th candidate is obtained as:

$$\widehat{m}^{(k)} = \text{Sigmoid}\left(\text{Conv}\left(\left[\mathbf{F}_{mask};\mathbf{F}^{(k)}_{pos};\mathbf{F}^{(k)}_{geo}\right];w^{(k)}\right)\right). \tag{3}$$

The geometric feature $\mathbf{F}^{(k)}_{geo} \in \mathbb{R}^{N\times 6}$ can be calculated from the absolute difference of the bounding box predicted by the k-th instance candidate and N points in the input point cloud $\mathbf{P}$, or $f^{(k,i)}_{geo} = \left|f^{(i)}_{box} - f^{(k)}_{box}\right|$.

3.3. Network Training

We train our approach with the *Pointwise Loss* and *Instance-wise Loss*. The former is incurred at the point-wise prediction, i.e., the cross-entropy loss for semantic segmentation, and the L1 loss and gIoU loss [31] for bounding box regression. The latter is incurred at each instance prediction namely classification, box prediction, and mask prediction using the one-to-many matching loss proposed by [20] for 2D object detection. Specifically, the matching cost is the combination of instance classification and instance masks:

$$C(k,j) = \gamma_{mask} C_{mask}(\widehat{m}^{(k)}, m^{(j)}) + C_{cls}(\widehat{l}^{(k)}, l^{(j)}), \tag{4}$$

where C_{mask} is the dice loss [32] between two masks. Precisely, S predicted masks are matched to one ground-truth mask by duplicating the ground truth S times in Hungarian matching. In this way, the training convergence is much faster and the mask prediction performance is better than the one-to-one matching proposed by DETR [3]. Then the instance-wise loss incurred between the ground-truth masks and their matched predicted masks is defined as:

$$L_{inst} = L_{cls} + \lambda_{box} L_{box} + \lambda_{mask} L_{mask} + \lambda_{ms} L_{MS}, \tag{5}$$

where L_{cls} is the cross-entropy loss, L_{mask} is the combination of dice loss and BCE loss, L_{box} is the combination of L1 loss and gIoU loss, and L_{MS} is the Mask-Scoring loss [19].

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on three datasets: ScanNetV2 [8], S3DIS [1], and STPLS3D [4]. The *ScanNetV2* dataset consists of 1201, 312, and 100 scans with 18 object classes for training, validation, and testing, respectively. We report the evaluation results on the validation and test sets of ScanNetV2 as in the previous work. The *S3DIS* dataset contains 271 scenes from 6 areas with 13 categories. We report evaluations for both Area 5 and 6-fold cross-validation. The *STPLS3D* dataset is an aerial photogrammetry point cloud dataset from real-world and synthetic environments. It includes 25 urban scenes of a total of 6km^2 and 14 instance categories. Following [5, 34], we use scenes 5, 10, 15, 20, and 25 for validation and the rest for training.

Evaluation Metrics. Average precision commonly used for object detection and instance segmentation tasks is adopted, i.e., AP$_{50}$ and AP$_{25}$ are the scores with IoU thresholds of 50% and 25%, respectively, while AP is the averaged scores with IoU thresholds from 50% to 95% with a step size of 5%. Box AP means the average precision of the 3D axis-aligned bounding box prediction. Additionally, the S3DIS is also evaluated using mean coverage (mCov), mean weighed coverage (mWCov), mean precision (mPrec$_{50}$), and mean recall (mRec$_{50}$) with IoU threshold of 50%.

Implementation Details. We implement our model using PyTorch deep learning framework [27] and train it on 320 epochs with AdamW optimizer on a single V100 GPU. The batch size is set to 16. The learning rate is initialized to 0.004 and scheduled by a cosine annealing [41]. Following [34], we set the voxel size to 0.02m for ScanNetV2 and S3DIS, and to 0.3m for STPLS3D due to its sparsity and much larger scale. In training, the scenes are randomly cropped at a maximum number of 250,000 points. In testing, the whole scenes are fed into the network without cropping. We use the same backbone design as in [34], which outputs a feature map of 32 channels. A stack of two layers of PA is used in the sampling-based instance-aware encoder. τ is set to 0.5. We set the ball query radius r to 0.2 and 0.4 for these two layers and the number of neighbors $Q = 32$ for both layers. We also implement the box-aware dynamic convolution with two layers with a hidden dimension of 32. γ_{mask} is set to 5. λ_{box}, λ_{mask}, and λ_{ms} are set to 1, 5, and 1, respectively. In training, we set $S = 4$ and $K = 256$. In inference, we set $K = 384$ and use Non-Maximum-Suppression to remove redundant mask predictions with a threshold of 0.2. Following [14, 26, 38], we leverage the superpoints [23, 24] to align the final predicted masks on the ScanNetV2 dataset.

4.2. Main Results

ScanNetV2. We report the instance segmentation results on the hidden test set in Tab. 2 and the instance segmentation

Method	**Venue**	**AP**	**AP$_{50}$**	**AP$_{25}$**
SGPN [35]	CVPR 18	4.9	14.3	26.1
MTML [39]	ICCV 19	28.2	54.9	73.1
3D-BoNet [39]	NeurIPS 19	25.3	48.8	68.7
PointGroup [21]	CVPR 20	40.7	63.6	77.8
OccuSeg [14]	CVPR 20	44.3	67.2	74.2
DyCo3D [16]	CVPR 21	39.5	64.1	76.1
PE [41]	CVPR 21	39.6	64.5	77.6
HAIS [5]	ICCV 21	45.7	69.9	80.3
SSTNet [26]	ICCV 21	50.6	69.8	78.9
SoftGroup [34]	CVPR 22	50.4	<u>76.1</u>	**86.5**
RPGN [9]	ECCV 22	42.8	64.3	80.6
PointInst3D [17]	ECCV 22	43.8	-	-
Di&Co3D [42]	ECCV 22	47.7	70.0	80.2
DKNet [38]	ECCV 22	<u>53.2</u>	71.8	81.5
ISBNet	-	**55.9**	**76.3**	<u>84.5</u>

Table 2. 3D instance segmentation results on ScanNetV2 hidden test set in terms of AP scores. The best results are in **bold** and the second best ones are in <u>underlined</u>. Our proposed method achieves the highest AP, outperforming the previous strongest method.

and object detection results on the validation set in Tab. 3. On the hidden test set, ISBNet achieves 55.9/76.6 in AP/AP$_{50}$, set a new state-of-the-art performance on ScanNetV2 benchmark. On the validation set, our proposed method surpasses the second-best method with large margins, $+3.7/5.5/3.6$ in AP/AP$_{50}$/AP$_{25}$ and $+2.6/6.5$ in Box AP$_{50}$/Box AP$_{25}$.

S3DIS. Tab. 4 summarizes the results on Area 5 and 6-fold cross-validation of the S3DIS dataset. On both Area 5 and cross-validation evaluations, our proposed method overtakes the strongest method by large margins in almost metrics. On the 6-fold cross-validation evaluation, we achieve 74.9/76.8/77.1 in mCov/mWCov/mRec$_{50}$, with an improvement of $+3.5/2.7/3.1$ compared with the second-strongest method.

STPLS3D. Tab. 5 shows the results on the validation set of the STPLS3D dataset. Our method achieves the highest performance in all metrics and surpasses the second-best by $+3.0/2.2$ in AP/AP$_{50}$.

4.3. Qualitative Results

We visualize the qualitative results of our method, DyCo3D [16], and DKNet [38] on ScanNetV2 validation set in Fig. 3. As can be seen, our method successfully distinguishes nearby instances with the same semantic class. Due to the limitation of clustering, DyCo3D [16] mis-segments parts of the bookshelf (row 1) and merges nearby sofas (rows 2, 3). DKNet [38] over-segments the window in row 2, and also wrongly merges nearby sofas and table (row 3).

4.4. Ablation Study

We conduct a series of ablation studies on the validation set of the ScanNetV2 dataset to investigate ISBNet.

Method	**AP**	**AP$_{50}$**	**AP$_{25}$**	**Box AP$_{50}$**	**Box AP$_{25}$**
GSPN [40]	19.3	37.8	53.4	10.8	19.8
PointGroup [21]	34.8	51.7	71.3	48.9	61.5
HAIS [5]	43.5	64.4	75.6	53.1	64.3
DyCo3D [16]	40.6	61.0	-	45.3	58.9
SSTNet [26]	49.4	64.3	74.0	52.7	62.5
SoftGroup [34]	46.0	<u>67.6</u>	<u>78.9</u>	<u>59.4</u>	<u>71.6</u>
RPGN [9]	-	64.2	-	-	-
PointInst3D [17]	45.6	63.7	-	51.0	-
Di&Co3D [42]	47.7	67.2	77.2	-	-
DKNet [38]	<u>50.8</u>	66.7	76.9	59.0	67.4
ISBNet	**54.5**	**73.1**	**82.5**	**62.0**	**78.1**

Table 3. 3D instance segmentation and 3D object detection results on ScanNetV2 validation set.

Method	**AP**	**AP$_{50}$**	**mCov**	**mWCov**	**mPrec$_{50}$**	**mRec$_{50}$**
SGPN† [40]	-	-	32.7	35.5	36.0	28.7
PointGroup† [21]	-	57.8	-	-	61.9	62.1
HAIS† [5]	-	-	64.3	66.0	71.1	65.0
SSTNet† [26]	42.7	59.3	-	-	65.6	64.2
SoftGroup† [34]	51.6	<u>66.1</u>	<u>66.1</u>	<u>68.0</u>	**73.6**	66.6
RPGN† [9]	-	-	-	-	64.0	63.0
PointInst3D† [17]	-	-	64.3	65.3	<u>73.1</u>	65.2
Di&Co3D† [42]	-	-	65.5	66.1	63.9	<u>67.2</u>
DKNet† [38]	-	-	64.7	65.6	70.8	65.3
ISBNet†	**56.3**	**67.5**	**70.0**	**70.7**	70.5	**72.0**
SGPN‡ [40]	-	54.4	37.9	40.8	38.2	31.2
3D-BoNet‡ [39]	-	-	-	-	65.6	47.7
PointGroup‡ [21]	-	64.0	-	-	69.6	69.2
OccuSeg‡ [14]	-	-	-	-	72.8	60.3
HAIS‡ [5]	-	-	67.0	70.4	73.2	69.4
SSTNet‡ [26]	54.1	67.8	-	-	73.5	73.4
SoftGroup‡ [34]	54.4	68.9	69.3	71.7	75.3	69.8
RPGN‡ [9]	-	-	-	-	**84.5**	70.5
PointInst3D‡ [17]	-	-	<u>71.5</u>	<u>74.1</u>	76.4	<u>74.0</u>
DKNet‡ [38]	-	-	70.3	72.8	75.3	71.1
ISBNet‡	**60.8**	**70.5**	**74.9**	**76.8**	<u>77.5</u>	**77.1**

Table 4. 3D instance segmentation results on S3DIS dataset. Methods marked with † are evaluated on Area 5, and methods marked with ‡ are evaluated on 6-fold cross-validation.

	AP	AP$_{50}$
PointGroup[21]	23.3	38.5
HAIS[5]	35.1	46.7
SoftGroup[34]	<u>46.2</u>	<u>61.8</u>
ISBNet	**49.2**	**64.0**

Table 5. 3D instance segmentation results on STPLS3D validation set.

BCE	Focal	Dice	AP	AP$_{50}$
✓			53.6	72.1
	✓		46.5	63.4
	✓	✓	53.9	72.1
✓		✓	**54.5**	**73.1**

Table 6. Ablation study on different combinations of mask losses on ScanNetV2 validation set.

The impact of different combinations of mask losses is

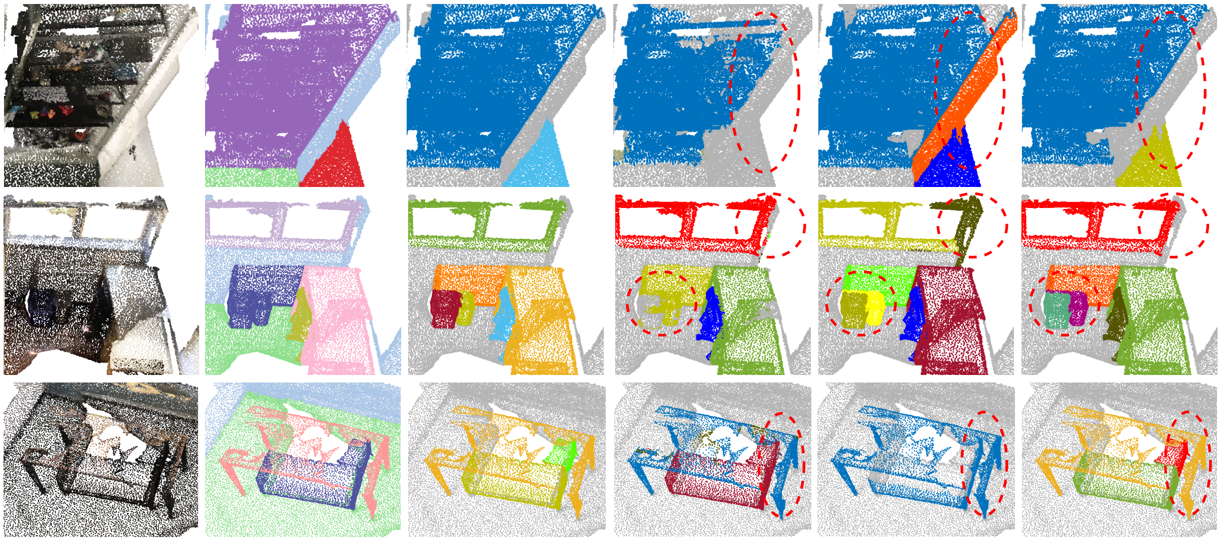

Figure 3. Representative examples on ScanNetV2 validation set. Each row shows an example with the input, Semantic ground truth, and Instance ground truth in the first three columns. Our method (the last column) produces more precise instance masks, especially in regions where multiple instances with the same semantic label lie together.

shown in Tab. 6. Notably, using a combination of binary cross entropy and dice loss yields the best result, 54.5 in AP.

The impact of each component on the overall performance is shown in Tab. 7. DyCo3D* in row 1 is our re-implementation of DyCo3D with the same backbone as [5, 34, 38], and it is trained with the one-to-many matching loss. The baseline in row 2 is a model with standard Farthest Point Sampling (FPS), standard Dynamic Convolution as in [16, 17, 38], and without Local Aggregation Layer (LAL) . It can be seen that replacing the clustering and the tiny Unet in DyCo3D* decreases the performance from 49.4 to 47.9 in AP. When the standard FPS in the baseline is replaced by the Instance-aware Farthest Point Sampling (IA-FPS), the performance improves to 49.7 in row 3. When adding LAL to the baseline model, the AP score increases to 50.1 in row 4 and outperforms the AP of DyCo3D* by 0.7. Simply replacing the standard Dynamic Convolution with the Box-aware Dynamic Convolution (BA-DyCo) gains +0.7 in AP in row 5. Especially, combining the IA-FPS and PA significantly boosts the performance, +5.5/+6.8 in AP/AP_{50} in row 6. Finally, the full approach in row 7, ISBNet achieves the best performance 54.5/73.1 in AP/AP_{50}.

The impact of axis-aligned bounding box regression is shown in Tab. 8. Without using the bounding box as an auxiliary supervision (**Aux.**), our method achieves 52.8/71.6 in AP/AP_{50}. Adding the bounding box loss during training brings a 0.6 improvement in AP. Especially, when using the bounding box as an extra geometric cue (**Geo. Cue**) in the dynamic convolution, the result significantly increases to 54.5/73.1 in AP/AP_{50}. This justifies our claim that the 3D bounding box is a critical geometric cue to distinguish instances in 3D point cloud.

	IA-FPS	LAL	BA-DyCo	AP	AP_{50}	AP_{25}
DyCo3D*				49.4	67.6	77.4
Baseline				47.9	66.4	77.1
	✓			49.7	67.5	78.6
		✓		50.1	69.4	79.1
			✓	48.6	67.7	77.8
	✓	✓		53.4	71.9	81.8
ISBNet	✓	✓	✓	**54.5**	**73.1**	**82.5**

Table 7. Impact of each component of ISBNet on ScanNetV2 validation set. **IA-FPS**: Instance-aware Farthest Point Sampling, **LAL**: Local Aggregation Layer, **BA-DyCo**: Box-aware Dynamic Convolution. *: our improved version of DyCo3D [16].

Aux.	Geo. Cue	AP	AP_{50}
		52.8	71.6
✓		53.4	71.9
✓	✓	**54.5**	**73.1**

Table 8. The impact of 3D axis-aligned bounding box regression.

# of PA	AP	AP_{50}
1	53.2	72.5
2	**54.5**	**73.1**
3	54.3	73.0

Table 9. The number of Point Aggregator (**PA**) blocks.

The impact of the number of Point Aggregator (PA) blocks is represented in Tab. 9. With a single block of PA, our method achieves 53.2/72.5 in AP/AP_{50}. Stacking two blocks of PA gives 1.3/0.6 gains in these metrics. However, when we add more blocks, the results slightly decrease to 54.3/73.0 in AP/AP_{50}.

The impact of different designs of the Dynamic Convolution is shown in Tab. 10. Here, using two layers of dynamic

# of layers	Dimensions	# of params	AP	AP$_{50}$
1	(41,1)	**41**	45.7	67.1
2	(25,8,1)	216	53.6	72.1
2	(41,16,1)	688	53.9	72.3
2	(41,32,1)	1376	**54.5**	**73.1**
3	(41,16,16,1)	960	53.9	72.7
3	(41,32,16,1)	1696	54.2	72.8

Table 10. Ablation on the Box-aware Dynamic Convolution.

Chunk size	Total samples K	AP	AP$_{50}$	AP$_{25}$
(256)	256	53.9	72.2	80.8
(384)	384	54.2	72.4	81.4
(512)	512	53.6	71.9	81.1
(128,128,128)	384	54.0	72.8	81.0
(192,128,64)	384	**54.5**	**73.1**	**82.5**

Table 11. Ablation on the sample chunk size of Iterative sampling.

convolution with the hidden channels of 32 gives the best results. Using only a single layer of dynamic convolution leads to a significant drop in performance. On the other hand, adding too many layers, i.e., three layers yields worse results. Reducing the number of hidden channels slightly decreases the performance. Thanks to the additional geometric cue, even with only 216 parameters of dynamic convolution, our model can achieve 53.6/72.1 in AP/AP$_{50}$, demonstrating the robustness of the box-aware dynamic convolution.

Ablation on the chunk size of IA-FPS. We study different designs of the sampling chunk size of IA-FPS in inference in Tab. 11. The first three rows show the results when we sample K candidates at once. Increasing the number of samples from 256 to 384 slightly improves the overall performance, but at 512 samples, the result drops to 53.6 in AP. When splitting K into smaller chunks of size (192,128,64) and sampling points based on Eq. (1), the performance further boosts to 54.5/73.1 in AP/AP$_{50}$ in the last row.

Runtime Analysis. Fig. 4 reports the component and total runtimes of ISBNet and 5 recent state-of-the-art methods of 3DIS on the same Titan X GPU. All the methods can be roughly separated into three main stages: backbone, instance abstractor, and mask decoder. Our method is the fastest method, with only 237ms in total runtime and 152/53/32 ms in backbone/instance abstractor/mask decoder stages. Compared with the instance abstractors in PointGroup [21], DyCo3D [16], and SoftGroup [34] which are based on clustering, our instance abstractor based on our Point Aggregator significantly reduce the runtime. Our mask decoder, which is implemented by dynamic convolution, is the second fastest among these methods. This proves the efficiency of our proposed method.

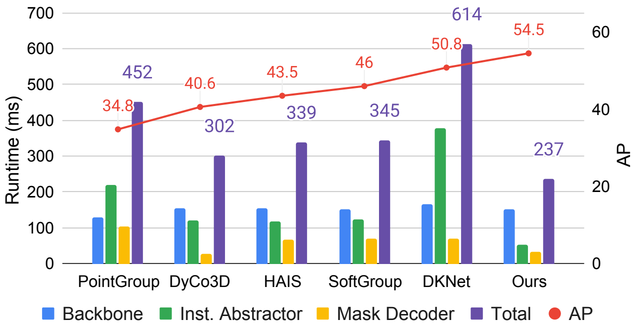

Figure 4. Components and total runtimes (in ms) and results in AP of five previous methods and ISBNet on ScanNetV2 validation set.

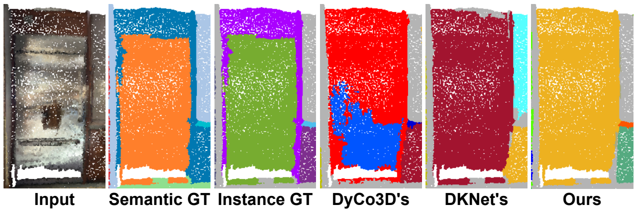

Figure 5. A hard case on ScanNetV2 validation set where a fridge is bounded by a counter. ISBNet and previous methods wrongly merge points from these instances into a single object.

5. Conclusion

In this work, we have introduced the ISBNet , a concise dynamic convolution-based approach to address the task of 3D point cloud instance segmentation. Considering the performance of instance segmentation models relying on the recall of candidate queries, we propose our Instance-aware Farthest Point Sampling and Point Aggregator to efficiently sample candidates in the 3D point cloud. Additionally, leveraging the 3D bounding box as auxiliary supervision and a geometric cue for dynamic convolution further enhances the accuracy of our model. Extensive experiments on ScanNetV2, S3DIS, and STPLS3D datasets show that our approach achieves robust and significant performance gain on all datasets, surpassing state-of-the-art approaches in 3D instance segmentation by large margins, i.e., +2.7, +2.4, +3.0 in AP on ScanNetV2, S3DIS and STPLS3D.

Our method is not without limitations. For example, our instance-aware FPS does not guarantee to cover all instances as it relies on the current instance prediction to make decisions for point sampling. Our proposed axis-aligned bounding box may not tightly fit the shape of complicated instances. A hard case is shown in Fig. 5 where a fridge is bounded by a counter. Our model cannot distinguish these points as they share similar bounding boxes. Addressing these limitations might lead to improvement in future work. Additionally, a new study on improving dynamic convolution by leveraging objects' geometric structures such as their shapes and sizes would be an interesting research topic.

References

[1] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[2] Z. Cai and N. Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[4] M. Chen, Q. Hu, Z. Yu, H. THOMAS, A. Feng, Y. Hou, K. McCullough, F. Ren, and L. Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. In *Proceedings of the British Machine Vision Conference*, 2022.

[5] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the International Conference on Computer Vision*, 2021.

[6] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, 2021.

[7] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] S. Dong, G. Lin, and T.-Y. Hung. Learning regional purity for instance segmentation on 3d point clouds. In *Proceedings of the European Conference on Computer Vision*, 2022.

[10] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.

[11] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[12] B. Graham, M. Engelcke, and L. Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[13] R. Guo, D. Niu, L. Qu, and Z. Li. Sotr: Segmenting objects with transformers. In *Proceedings of the International Conference on Computer Vision*, 2021.

[14] L. Han, T. Zheng, L. Xu, and L. Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the International Conference on Computer Vision*, 2017.

[16] T. He, C. Shen, and A. van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[17] T. He, C. Shen, and A. van den Hengel. Pointinst3d: Segmenting 3d instances by points. In *Proceedings of the European Conference on Computer Vision*, 2022.

[18] J. Hou, A. Dai, and M. Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[19] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang. Mask scoring r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] D. Jia, Y. Yuan, H. He, X. Wu, H. Yu, W. Lin, L. Sun, C. Zhang, and H. Hu. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022.

[21] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[22] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[23] L. Landrieu and M. Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[24] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[25] F. Li, H. Zhang, S. Liu, L. Zhang, L. M. Ni, H.-Y. Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022.

[26] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the International Conference on Computer Vision*, 2021.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[28] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[32] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J.

Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017.

[33] Z. Tian, C. Shen, and H. Chen. Conditional convolutions for instance segmentation. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020.

[34] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo. Softgroup for 3d instance segmentation on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[35] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li. Solo: Segmenting objects by locations. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020.

[37] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen. Solov2: Dynamic and fast instance segmentation. In *Advances in Neural Information Processing Systems*, 2020.

[38] Y. Wu, M. Shi, S. Du, H. Lu, Z. Cao, and W. Zhong. 3d instances as 1d kernels. In *Proceedings of the European Conference on Computer Vision*, 2022.

[39] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, 2019.

[40] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[41] B. Zhang and P. Wonka. Point cloud instance segmentation using probabilistic embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[42] W. Zhao, Y. Yan, C. Yang, J. Ye, X. Yang, and K. Huang. Divide and conquer: 3d point cloud instance segmentation with point-wise binarization. In *Proceedings of the European Conference on Computer Vision*, 2022.