DyCo3D: Robust Instance Segmentation of 3D Point Clouds through Dynamic Convolution

Tong He, Chunhua Shen[*], Anton van den Hengel
The University of Adelaide, Australia

Abstract

Previous top-performing approaches for point cloud instance segmentation involve a bottom-up strategy, which often includes inefficient operations or complex pipelines, such as grouping over-segmented components, introducing additional steps for refining, or designing complicated loss functions. The inevitable variation in the instance scales can lead bottom-up methods to become particularly sensitive to hyper-parameter values. To this end, we propose instead a dynamic, proposal-free, data-driven approach that generates the appropriate convolution kernels to apply in response to the nature of the instances. To make the kernels discriminative, we explore a large context by gathering homogeneous points that share identical semantic categories and have close votes for the geometric centroids. Instances are then decoded by several simple convolutional layers. Due to the limited receptive field introduced by the sparse convolution, a small light-weight transformer is also devised to capture the long-range dependencies and high-level interactions among point samples. The proposed method achieves promising results on both ScanetNetV2 and S3DIS, and this performance is robust to the particular hyper-parameter values chosen. It also improves inference speed by more than 25% over the current state-of-the-art. Code is available at: `https://git.io/DyCo3D`

1. Introduction

Instance segmentation is significantly more challenging than semantic segmentation because it requires identifying every individual instance of a class of objects, and the visible extent of each. The information recovered has proven invaluable for scene understanding, however. With the increasing applications of 3D sensors (such as LiDAR and laser scanners), point clouds have become an important modality in scene understanding. Although significant advances have been made in instance segmentation in the image domain [15, 4, 3, 36], instance segmentation with the 3D point clouds has proven far more challenging. This is partly due to the inherent irregularity and sparsity of the data, but also due to the diversity of the scene. By way of example, Mask R-CNN [15], which has shown great success when applied to 2D images, performs poorly when applied in 3D [19].

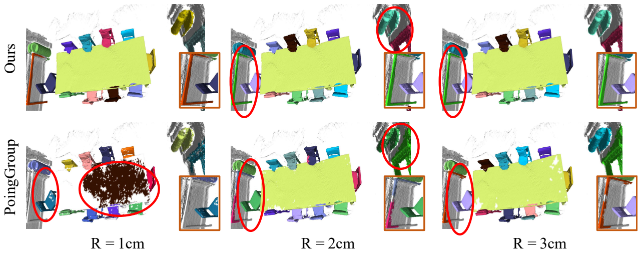


Figure 1 – A comparison of the instance segmentation results achieved using DyCo3D, and PointGroup [20]. Our method shows better robustness and generalization to the varying hyper-parameter values. Different instances are shown with random colors, and red ellipses highlight specific over-segmentation errors. Best viewed in color.

Many previous top-performing approaches for point cloud instance segmentation adopt a bottom-up strategy, involving heuristic grouping algorithms or complex post-processing steps. 3D-MPA [11], for example, extracts proposals from the predicted instance centroids. Instances are then generated by aggregating proposal-wise embeddings. PointGroup [20] generates instances proposals by gradually merging neighbouring points that share the same category label. Both original and centroid-shifted points are explored with a manually specified search radius. A separate model (labelled ScoreNet) is used to estimate the objectness of the proposals. Both methods achieved promising performance on ScanNetV2 [7] and S3DIS [1] benchmarks. However, these bottom-up methods often suffer from several drawbacks: (1) the performance is sensitive to values of the pre-defined hyper-parameters, which require manual tuning. In PointGroup [20], modifying the clustering radius

[*] Corresponding author (e-mail: `chunhua@icloud.com`).

from 3cm to 2cm causes mAP to drop by more than 6%, illustrating the method's limited robustness and generalization ability. These results are presented in Fig. 1. (2) they incorporate either complex post-processing steps or training pipelines, rendering them unsuitable for real-time applications such as robotics and driverless cars. For example, 3D-MPA [11] needs an extra 10-layer graph network and a clustering post-processing step to yield its final instance segmentation masks. (3) they are heavily reliant on the quality of the proposals, which limits their robustness and can lead to joint/fragmented instances in practice.

In this paper, we propose a novel pipeline tailored to 3D point cloud instance segmentation using dynamic convolution, that we label DyCo3D. Our approach addresses the task with only a few convolution layers, for which the filters are generated on the fly, conditioned on the category and position of the instance to be decoded. To empower the filters to distinguish different instances, we propose to encode category-specific context by deploying a light-weight sub-network to explore homogenous points that have close votes for instance centroids and share the semantic labels. Instance masks can be decoded in parallel by convolving the generated class-specific filters with the position embedded features. Compared with bottom-up approaches [20, 11, 39, 38, 17] that are sensitive to the values of numerous hyper-parameters, our approach demonstrates superiority on both effectiveness and efficiency. Qualitative results illustrating this fact are presented in Fig. 1.

Besides, as has been proved in the 2D image domain, a large receptive field and rich context information are critical to the success of instance segmentation [5]. To address the problem in the 3D point cloud, we propose to introduce a small transformer [37] to capture a long-range dependency and build high-level interactions among different regions.

Our contributions are summarised as the following:

- A novel method for 3D point cloud instance segmentation based on dynamic convolution that outperforms previous methods in both efficiency and effectiveness.
- A proposal-free instance segmentation approach that is more robust than bottom-up strategies.
- A light-weight transformer that enlarges the receptive field and captures non-local dependencies.
- Comprehensive experiments demonstrating that the proposed method achieves state-of-the-art results, with improved robustness, and an inference speed superior to that of its comparators.

2. Related Work

Deep Learning for 3D Point Cloud. In contrast to the image domain, wherein the data representation is relatively consistent (see *e.g.* VGGNet [32] and ResNet [16]), methods for 3D point cloud representation are still developing. The most prevalent existing approaches can be roughly categorised as: point-based [29, 31], voxel-based [23, 40, 13, 6], and multiview-based [35, 30, 8]. PointNet [29] is one of the pioneering point-based approaches. It exploits a shareable multi-layer perceptron (MLP) network to extract per-point representation. PointNet++ [31] extends this approach by introducing a nested hierarchical PointNet to extract local context information. Although simple, PointNet and PointNet++ are still widely used in the tasks of semantic segmentation [31, 41, 27], instance segmentation [39, 38, 18, 17], and 3D detection [43, 26]. Multi-view solutions often involve view projection to utilize well-explored 2D techniques. In [35], for instance, view-pooling is used to combine information from different views of a 3D shape and thereby to construct a compact shape descriptor. 3DSIS [19], in contrast, projects features extracted from 2D views into 3D space. Voxel-based methods first transfer 3D points into rasterized voxels and apply convolution operations for feature extraction. Traditional 3D convolution methods [23, 34] are often constrained by inefficient computation and limited GPU memory. In addition, computational and representational resources are wasted on void space. DyCo3D, in contrast, uses sparse volumetric convolution [13, 6] to efficiently process this inherently sparse data. Focusing computation on the data, rather than the space it occupies, makes DyCo3D faster, more robust, and better able to extract local patterns.

Instance Segmentation of 3D Point Cloud. As in the 2D image domain, 3D instance segmentation approaches can be broadly divided into two groups: top-down and bottom-up. Top-down methods often use a detect-then-segment approach, which first detects 3D bounding boxes of the instances and then predicts foreground points. 3D-BoNet [42], for instance, first detects unoriented 3D bounding boxes from a single global representation by utilizing a Hungarian matching algorithm. Then per-point features are explored within each bounding box to mask out the background. Instead of regressing bounding boxes for instance proposals, GSPN [44] generates instance shapes and applies analysis-by-synthesis. Bottom-up methods, in contrast, group sub-components into instances. Methods applying this approach have dominated the leaderboard of the ScanNet dataset [7][1]. The grouping techniques vary from simple clustering [14, 39, 17, 25, 18, 20] to complex graph-based algorithms [11, 14] based on learned embeddings. ASIS [39], for example, learns point-level embeddings, regularized by a discriminative loss function [2], which encourages points belonging to the same instance to be mapped to similar locations in a metric space while separating points belonging to different instances. A mean-shift algorithm is then applied to generate instance masks. Many subse-

[1] http://kaldir.vc.in.tum.de/scannet_benchmark/

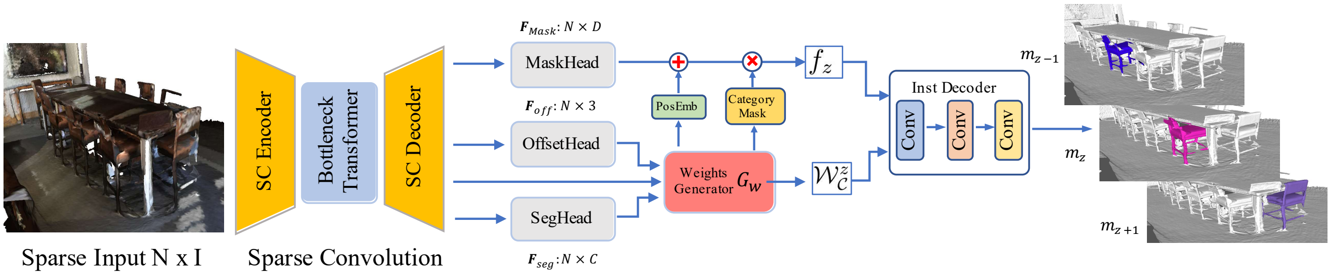


Figure 2 – The structure of DyCo3D. It contains three main components: (1) a sparse convolution backbone based on [13], which contains a light-weight transformer and outputs three parallel heads for instance mask generation, offset prediction, and semantic segmentation. (2) A weight generator that takes centroid predictions and semantic segmentations as input. Homogenous points that have close votes for instance centroids and share the category predictions are explored to output instance-aware position embeddings, category-specific masks, and convolutional filters. (3) An instance decoder. Binary masks of instances are decoded by applying several convolutions, with the filters constructed by the Weight Generator.

quent works [18, 25, 17, 45] use the same general pipeline. PointGroup [20] generates instance clusters from two sets of points: original and centroid shifted points. A network the authors label ScoreNet is used to evaluate the candidates. OccuSeg [14], in contrast, uses multi-task learning to generate feature embeddings, but also explicit occupancy embeddings that enable metric instance scale calculations to be made.

Dynamic Convolution. The existing works most closely related to DyCo3D are [9] and [36]. Dynamic convolution was first proposed to enhance filter representation by encoding sample-specific and position-specific knowledge. CondInst [36] successfully applies it in the 2D image domain for instance segmentation. However, our experiments demonstrate that it performs poorly when applied directly to 3D point clouds for the following reasons: (1) it introduces a large amount of computation, resulting in optimization difficulties. (2) the performance is constrained by the limited receptive field and representation capability due to the sparse convolution. In this paper, we improve the dynamic convolution tailored for 3D point cloud instance segmentation and demonstrate its effectiveness and robustness on multiple benchmarks.

3. Methods

3.1. Overall Architecture

The structure of DyCo3D is depicted in Fig. 2. The input to the network is a matrix recording the point features $\mathbf{P} \in \mathbb{R}^{N \times I}$, where N is the total number of points and I is the dimension of each point feature. The goal is to predict a set of point-level binary masks and their corresponding category labels, denoted as $\{(\hat{m}_k, \hat{c}_k)\}$, where $\hat{m}_k \in \{0, 1\}^N$, and $\hat{c}_k \in \{1, 2, \cdots, C\}$. C is 20 for ScanNetV2 [7] and 13 for S3DIS [1]. Compared with previous top-performing approaches [20, 11], where instances masks are dependent on the proposals, our method is proposal-free and can produce instance masks using only a small number of simple convolutional layers. The associated convolution filters are dynamically generated, conditioned on both spatial distribution of the data and the semantic predictions. As shown in Fig. 2, DyCo3D is comprised of three primary components: (1) a backbone network, which is based on sparse convolution for feature extraction, and contains a light-weighted transformer [37], aiming to enlarge the receptive field and capture long-range dependencies. (2) A weight generator that responds to the individual characteristics of each instance to dynamically generate the appropriate filter parameters. To make the filters discriminative, a large category-specific context is introduced. (3) An instance decoder. Instances are separated in parallel, using only three convolution layers, by convolving the generated class-aware filters with position embedded features.

3.2. Backbone Network

Although our method is not restricted to any specific choice of backbones, we select sparse convolution [13] for its efficiency and competitive performance. Following [13, 20], we construct a U-Net, which consists of an encoder and a decoder that have symmetrical structures. However, sparse convolution is often constrained by a limited receptive field and representation capability, due to the small number of convolution layers and channels. To this end, we propose a light-weight transformer [37] to enhance long-range interactions on top of the encoder. The transformer is identical to the implementation of [37], except for the position embedding layer, where the position-sensitive information is encoded as the mean of the pairwise direction vector or relative position.

We denote the features output by the backbone as $\mathbf{F}_b \in \mathbb{R}^{N \times D}$, where D is the dimension of the output channel. Three parallel branches are built upon $\mathbf{F}_b$ for semantic segmentation ($\mathbf{F}_{\text{seg}} \in \mathbb{R}^{N\times C}$), offset prediction ($\mathbf{O}_{\text{off}} \in \mathbb{R}^{N\times 3}$), and instance masking ($\mathbf{F}_{\text{mask}} \in \mathbb{R}^{N\times D}$), where C is the category number.

Semantic Segmentation. We apply traditional cross entropy loss $\mathcal{L}_{\text{seg}}$ for semantic segmentation. Pointwise prediction of the category label can be easily obtained, indicated as $\{l^i_{\text{seg}}\}^N_{\text{i=1}}$.

Centroid Offset. The variability in the distribution of points across surfaces makes aggregating contextual information complex. To address the problem, we follow VoteNet [26], by shifting points towards the corresponding centroids of instances. Point-wise prediction o^i_{off} is supervised by the following loss function:

$$\mathcal{L}_{\text{ctr}} = \frac{1}{N_v} \sum_{i=0}^{N} \|p^i + o^i_{\text{off}} - ctr^i_{\text{gt}}\| \cdot \mathbb{1}(p^i) \qquad (1)$$

where p^i is the coordinates of the i-th point, o^i_{off} is the i-th item of $\mathbf{O}_{\text{off}}$, and ctr^i_{gt} is the geometric centroid of the corresponding instance. $\mathbb{1}(p^i)$ is an indicator function, representing whether p^i is a valid point for centroid prediction. N_v is the total number of the valid points. For example, the categories of 'floor' and 'wall' are ignored for instance segmentation on ScanNetV2 [7], making them free from offset predictions.

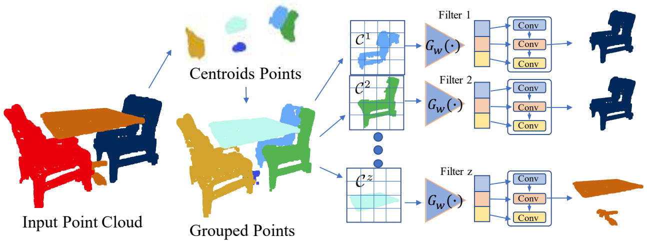


Figure 3 – The pipeline of the weight generator. Homogenous points are clustered by exploring both category prediction and geometric distribution. A light-weight sub-network is then applied to incorporate the larger context and applied once for each cluster to generate the convolution parameters used in instance decoding. Each filter is responsible for one instance.

3.3. Dynamic Weight Generator

The combination of the shallow network architecture and sparse convolution would typically cause a limited receptive field, and impair the method's ability to exploit large-scale context. To generate discriminative filters for distinguishing different instances we propose to group homogenous points that have close votes for the geometric centroids and share the category predictions. Then instance-aware filters are dynamically generated by applying a small sub-network for large context aggregation, as shown in Fig. 3. Provided both predicted semantic labels and centroids offsets, we are ready for grouping homogenous points by using a similar strategy to that in [20]. However, different from [20] that directly treats the clusters as individual instance proposals, our method explores the spatial distribution of these points and integrates large context to generate filters for instances decoding. Due to the removal of the reliance on the quality of the instance proposals, the performance of our method is robust to the pre-defined hyper-parameters, as approved in the following experiments. Qualitative results are presented in Fig. 1. Moreover, compared with CondInst [36], where filters are generated for every valid pixel, DyCo3D generates much less number of instance candidates (less than 60), and each filter is responsible for one instance in a specific class, reducing the difficulties for optimization and the heavy requirements for hardware resources. Given point-wise offset prediction $\{o^i_{\text{off}}\}^N_{\text{i=1}}$, centroids distribution $\{p^i_{\text{ctr}} \in \mathbb{R}^3\}^N_{\text{i=1}}$ can be easily calculated by $p^i_{\text{ctr}} = p^i + o^i_{\text{off}}$. With $\{p^i_{\text{ctr}}\}^N_{i=1}$ and semantic labels $\{l^i_{\text{seg}}\}^N_{i=1}$, instances are separated to a certain extent. We explore the void spaces among instances by applying a breadth-first searching algorithm [20] to group homogenous points that have identical semantic labels and close centroids predictions. Point p^j can be grouped with p^i if it satisfies: (1) $l^j_{\text{seg}} = l^i_{\text{seg}}$. (2) $\|p^j_{\text{ctr}} - p^i_{\text{ctr}}\|_2 <= r$, where r is a pre-defined searching radius. The grouping process ends up with a set of clusters $\{\mathcal{C}^z\}^Z_{\text{z=1}}$, where Z refers to the total number of clusters. As only one specific category is considered for each cluster, semantic label $l^z_{\mathcal{C}} \in \mathbb{R}$ of cluster $\mathcal{C}^z$ can be easily obtained from the semantic prediction. We also label the geometrical centroids for cluster $\mathcal{C}^z$ as $\mathcal{C}^z_{\text{ctr}} \in \mathbb{R}^3$, which is calculated as the average of the coordinates of the points in $\mathcal{C}^z$. Each cluster contains a bunch of points that are distributed across the instance, introducing a large context and rich geometric information. We explore the clusters and generate instance-aware weights for responding to the individual characteristics of each instance.

For cluster $\mathcal{C}^z$, we first voxelize it with a grid size of g, which is set to 14 in all our experiments. The features of each grid is calculated as the average of the point feature $\mathbf{F}_b$ within the grid, where $\mathbf{F}_b$ is the output of the backbone. To aggregate context for cluster $\mathcal{C}^z$, a light-weighted sub-network $G_w(\cdot)$ is maintained. It contains two sparse convolutional layers with a kernel size of 3, a global pooling layer, and an MLP layer. The output is all convolutional parameters flattened in a compact vector, $\mathcal{W}^z_{\mathcal{C}}$. Each $\mathcal{W}^z_{\mathcal{C}}$ is responsible for one specific instance. The size of $\mathcal{W}^z_{\mathcal{C}}$ is decided by the feature dimension and the number of the subsequent convolution layers (see Eq. 3).

3.4. Instance Decoder

Given a specific category, position representation is critical to separate different instances. To encode position sensitive knowledge, we directly append position embeddings in the feature space. For z-th instance with geometric centroid of $\mathcal{C}^z_{\text{ctr}}$, position embedding for the i-th point f^i_{pos} is calculated as:

$$f^i_{\text{pos}} = p^i - \mathcal{C}^z_{\text{ctr}} \tag{2}$$

where p^i is the coordinates of the i-th point. For $\mathcal{W}^z_{\mathcal{C}}$, the input feature $\{f^i_z \in \mathbb{R}^{D+3}\}^N_{i=1}$ is generated by concatenating $\{f^i_{\text{pos}} \in \mathbb{R}^3\}^N_{i=1}$ and $\{f^i_{\text{mask}} \in \mathbb{R}^D\}^N_{i=1}$.

Provided both instance-aware filters $\{\mathcal{W}^z_{\mathcal{C}}\}^Z_{z=1}$ and position-embedded features $\{f_z \in \mathbb{R}^{N\times(D+3)}\}^Z_{z=1}$, we are ready to decode binary segmentations of instances. The whole decoder contains three convolution layers with a kernel size of 1×1. Each layer uses ReLU as the activation function without normalization. Supposing the feature dimension of f^i_{mask} is 8, meaning $D = 8$, and the feature dimension of the decoder is 8, the total number of parameters (including both weights and biases) of $\mathcal{W}^z_{\mathcal{C}}$ is 177, which is calculated by:

$$177 = \underbrace{(8+3)\times 8 + 8}_{conv1} + \underbrace{8\times 8 + 8}_{conv2} + \underbrace{8\times 1 + 1}_{conv3} \tag{3}$$

Formally, the instance decoder is formulated as:

$$m_z = Conv(\mathcal{W}^z_{\mathcal{C}}, f_z) \tag{4}$$

where $m_z \in \mathbb{R}^N$ is the predicted binary mask for the z-th instance. Also, as filters are derived from a set of points that have identical semantic labels, we propose to operate the convolution on the points that have the same semantic predictions with $l^z_{\mathcal{C}}$. During training, the ground truth for $\mathcal{C}^z$ is $\hat{m}_z$ if it has the largest number of points in $\mathcal{C}^z$. The loss function for instance segmentation is defined as:

$$\mathcal{L}_{\text{mask}} = \frac{1}{Z}\sum^{Z}_{z=1}\frac{1}{N_z}\sum^{N}_{j=1}\mathbb{1}_{l^j_{\text{seg}}=l^z_{\mathcal{C}}} \cdot L_{\text{BCE}}(m^j_z, \hat{m}^j_z) \tag{5}$$

where Z is the total number of the clusters, l^j_{seg} is the semantic prediction of the j-th point, $l^z_{\mathcal{C}}$ is the semantic label of the z-th cluster, and L_{BCE} is the binary cross entropy loss function. $\mathbb{1}$ is an indicator function, showing the loss is only computed on the points that have identical semantic labels with group $\mathcal{C}^z$, and N_z is a normalization item which is calculated as: $\sum^N_{j=1}\mathbb{1}_{l^j_{\text{seg}}=l^z_{\mathcal{C}}}$. In addition to the point-wise supervision, we also utilize the dice loss [36] $\mathcal{L}_{\text{dice}}$, which is designed for addressing the imbalance between the foreground and background points.

3.5. Training details

The loss function of DyCo3D can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{ctr}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{dice}} \tag{6}$$

where $\mathcal{L}_{\text{seg}}$ is for semantic segmentation, $\mathcal{L}_{\text{ctr}}$ is for instance centroids supervision, and $\mathcal{L}_{\text{mask}}$ and $\mathcal{L}_{\text{dice}}$ are two loss items for instance segmentation. All loss weights are set to 1.0.

During the inference time, we perform NMS on the instance binary masks $\{m^z_{\mathcal{C}}\}^Z_{z=1}$, which are scored by the mean value of the semantic scores among the foreground points. The IoU threshold is the same as [20], with a value of 0.3. Cluster $\mathcal{C}^z$ is ignored if it contains points less than 50.

The voxel size is set to 0.02m and 0.05m for ScanNetV2 [7] and S3DIS [1], respectively. The hyper-parameter r of the searching radius is set to 0.03m, which is the same with [20] for a fair comparison. We implement multi-GPU training with a batch size of 16 and 4 GPUs. For the first 12k iterations, we only train the semantic segmentation $\mathcal{L}_{\text{seg}}$ and centroid prediction $\mathcal{L}_{\text{ctr}}$, as dynamic filters depend on the results of both tasks. For the next 38k iterations, we compute all the loss items. During the training, the initial learning rate is set to 0.01 with an Adam optimizer. We apply the same data augmentation strategy with [20], including random cropping, flipping, and rotating.

4. Experiments

To validate the effectiveness of our proposed method, we conduct both qualitative and quantitative experiments on datasets that are widely studied: ScanNetV2 [7] and Stanford 3D Indoor Semantic Dataset (S3DIS) [1]. In this section, we show that our method demonstrates superiority in both effectiveness and efficiency.

4.1. Datasets

S3DIS contains 13 categories that commonly exist in indoor scenes. The point cloud data is collected in 6 large-scale areas, covering more than 6000 m^2 with more than 215 million points. Following the protocols of previous methods [38, 17], we evaluate the performance on Area-5 and train the model on the other sets. ScanNet [7] is another large-scale benchmark for indoor scene analysis, which consists of 1613 scans with 40 categories in total. The dataset is split into 1201, 312, and 100 for training, evaluating, and testing, respectively. Like previous methods, we estimate the performance of instance segmentation on 18 common categories. Also, we follow the strategy in 3D-MAP [11] and report the performance of 3D detection, where the results are obtained by fitting an axis-aligned bounding box around the instance segmentation.

4.2. Evaluation Metrics

For ScanNetV2, we report the metric of mean average precision (mAP), which is widely used in the 2D image domain. AP@50 and AP@25 are also provided, having an IoU threshold set to 0.5 and 0.25, respectively. For S3DIS, we apply the metrics that are used in [39, 18, 17]: mConv,

Method	Group	PosEmb	CAD	TF	mAP	AP@50	AP@25
Baseline					24.8	43.8	56.4
CondInst		✓			27.0	44.7	57.5
	✓				29.4	49.7	66.3
	✓	✓			31.8	52.9	68.4
	✓	✓	✓		34.1	55.3	69.5
Ours-8	✓	✓	✓	✓	34.8	55.7	71.2
Ours-16	✓	✓	✓	✓	35.4	57.6	72.9

Table 1 – Ablation studies on the components of our proposed method. We evaluate the performance on the ScanNetV2 [7] validation set. **Group** indicates that the dynamic filters are generated by gathering homogenous points that share the semantic labels and have close centroids votes. **PosEmb** refers to the position embeddings f_{pos}. **CAD** denotes the category-aware decoding that the convolution in the decoding process is only operated on category-specific points, instead of all points. **TF** refers to the light-weight transformer applied for the backbone. With the same backbone, **ours-8** refers to the feature dimension of the mask head is 8, while **ours-16** denotes the dimension is 16.

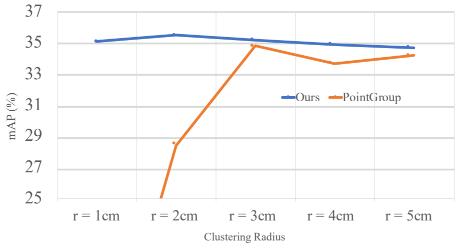


Figure 4 – The performance of the instance segmentation with different clustering radius r. All numbers for PointGroup are obtained from the paper or tested by the provided model. Unlike PointGroup, which is sensitive to the hyper-parameter and requires heuristic tuning, our method shows strong robustness.

mWConv, mPrec, and mRec. mConv is defined as the mean instance-wise IoU. mWConv denotes the weighted version of mConv, where the weights are determined by the sizes of instances. mPrec and mRec denote the mean precision and recall, respectively.

4.3. Ablation Studies

In this section, we analyze the effect of each component in our proposed DyCo3D. Performance is reported in terms of mAP, AP@50, and AP@25. All experiments are conducted with the same setting and training schedule, and are evaluated on ScanNetV2 [7] validation set.

Baseline. We build a strong baseline by generating filters for each foreground point without introducing any clustering operation. Due to the large size of N, we randomly select 150 points for instance decoding. As presented in Tab. 1, our method achieves 24.8, 43.8, and 56.4 in terms of mAP, AP@50, and AP@25, respectively. We also implement CondInst [36], which has demonstrated its success in the 2D image domain. As presented in the second row in Tab. 1 the mAP has boosted by 2.2%, with the help of instance-related position embeddings.

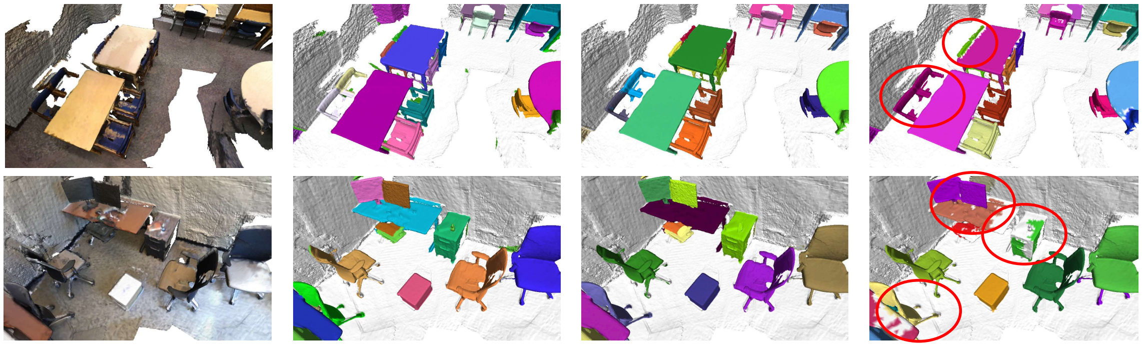
Ablation on the Grouping Homogenous Points. Due to the limited receptive field introduced by the sparse convolution, it is significant to incorporate rich context for distinguishing different instances. To this end, we propose to integrate homogenous points that are defined in Sec. 3.3. Thanks to the grouping operation, the model surpasses the baseline by a large margin in terms of all metrics. Besides, the grouping operation reduces the number of instance candidates (less than 60), lowering the optimization difficulties and the heavy requirements for the hardware facilities.

Ablation on the Category-Aware Decoding. As filters are generated by exploring the points that have identical semantic predictions, only certain category context is encoded. We propose to convolve each filter on these category-specific points and mask out irrelevant points. As presented in Tab. 1, adding category masks improves the mAP from 31.8% to 34.1%.

Ablation on the Transformer. As limited receptive field and representation ability introduced by the sparse convolution, we propose to add a light-weighted transformer upon the bottleneck layer to capture the long-range dependencies and enhance interactions among different points, while maintaining efficiency. As presented in Tab. 1, the transformer brings about 0.7% improvements in terms of mAP.

Ablation on the Clustering Radius. The clustering radius is pre-defined in the grouping step. PointGroup [20], which treats clusters as the instance proposals, makes the performance highly dependent on the quality of the clustering results. We test the performance with a different radius, as shown in Fig. 4. Grouping with a small r may generate over-segmented results, while a large r increases the risk of merging two adjacent objects. As a result, changing the radius r from 3cm to 2cm drops mAP by 6.3%, and 23.9% by changing r from 3cm to 1cm. The volatility makes it necessary to be carefully tuned, demonstrating limited generalization capability to various scenes. Our method, on the other hand, eliminates the dependence on the proposals, showing strong robustness to the radius r. More qualitative results can be found in Fig. 5.

Analysis on Efficiency. Different from previous point-based approaches that require to split each scene as 1m × 1m blocks and apply a complex block merging algorithm [38, 18, 17, 39], our method takes the whole scene as input. In addition, we also compare our DyCo3D with PointGroup, which has shown its efficiency on large-scale scenes. We report the inference time that is averaged on the whole validation set. With the only post-processing step



Input Point Cloud Ground Truth Ours PointGroup

Figure 5 – Comparison of the results with PointGroup [20]. The ellipses highlight specific over-segmentation/joint errors.

3D Object Detection		
ScanNetV2	AP@25%	AP@50%
DSS [33]	15.2	6.8
MRCNN 2D-3D [15]	17.3	10.5
F-PointNet [28]	19.8	10.8
GSPN [44]	30.6	17.7
3D-SIS [19]	40.2	22.5
VoteNet [26]	58.6	33.5
PointGroup [20]	56.8	42.3
Ours	**58.9**	**45.3**

Table 2 – 3D object detection on the validation set of ScanNetV2 [7]. We report per-class average precision (AP) with IoU thresholds of 25 % and 50 %. The performance of PointGroup [20] is evaluated with the provided model. We use the same backbone as [20] for a fair comparison.

Method	mCov	mWCov	mPrec	mRec
SGPN'18 [38]	32.7	35.5	36.0	28.7
ASIS'19 [39]	44.6	47.8	55.3	42.4
3D-BoNet'19 [42]	-	-	57.5	40.2
3D-MPA'20 [11]	-	-	63.1	58.0
MPNet'20 [17]	50.1	53.2	62.5	49.0
InsEmb'20 [18]	49.9	53.2	61.3	48.5
PointGroup'20 [20]	-	-	61.9	62.1
Ours	**63.5**	**64.6**	**64.3**	**64.2**

Table 3 – The results of instance segmentation on the S3DIS dataset. Performance on Area-5 is reported. A comparison with previous top-performing approaches is presented.

NMS, our method runs at 0.28s per scan on a 1080TI GPU, while the PointGroup runs at 0.39s with the same facility.

4.4. Comparison with State-of-the-art Methods

Object Detection. Following [11], we report the performance of 3D object detection on the validation set of ScanNetV2, which is obtained by fitting axis-aligned bounding boxes containing the instances. As shown in Tab. 2, our method surpasses PointGroup [20] by 3.1% and 3.0% in terms of AP@25 and AP@50, respectively, demonstrating the compactness of our generated instance masks.

Instance Segmentation on S3DIS. We report the performance of instance segmentation on the S3DIS benchmark, as shown in Tab. 3. Our method achieves the highest performance with all the evaluation metrics. The results in terms of mPrec and mRec are 2.6% and 2.1% higher than PointGroup [20]. Our method also reaches 60.9% under the metric of AP@50, which is 3.1% higher than PointGroup [20]. We compute all these metrics with the evaluation code provided by [39]. Qualitative results are illustrated in Fig. 6.

Instance Segmentation on ScanNetV2. We report the results of instance segmentation on the validation and testing sets of ScanNetV2, as presented in Tab. 4 and Tab. 5, respectively. We report both AP@50 and mAP on the validation set. We implement DyCo3D with both small and large backbones, denoted as **Ours-S** and **Ours-L**, respectively. Two models share the same network structure but with a different number of channels for convolution. We first compare **Ours-S** and PointGroup, which are implemented with the same backbone. Our method surpasses it by 0.7% and 0.6% in terms of AP@50 and mAP, respectively. We also make a fair comparison with 3D-MPA [11], our large model surpasses it by 2.3% and 4.9% in terms of AP@50 and mAP, respectively. We also report the performance of DyCo3D on the test set, as shown in Tab. 5. Highest AP@50 is achieved.

5. Conclusion

Achieving robustness to the inevitable variation in the data has been one of the ongoing challenges in 3D point cloud segmentation. We have shown here that dynamic

	AP@50	**mAP**	cabinet	bed	chair	sofa	table	door	window	bookshe.	picture	counter	desk	curtain	fridge	s.curtain	toilet	sink	bath	otherfu.
SegClu [19]	10.8	-	10.4	11.9	15.5	12.8	12.4	10.1	10.1	10.3	0.0	11.7	10.4	11.4	0.0	13.9	17.2	11.5	14.2	10.5
MRCNN [15]	9.1	-	11.2	10.6	10.6	11.4	10.8	10.3	0.0	0.0	11.1	10.1	0.0	10.0	12.8	0.0	18.9	13.1	11.8	11.6
SGPN [38]	11.3	-	10.1	16.4	20.2	20.7	14.7	11.1	11.1	0.0	0.0	10.0	10.3	12.8	0.0	0.0	48.7	16.5	0.0	0.0
3D-SIS [19]	18.7	-	19.7	37.7	40.5	31.9	15.9	18.1	0.0	11.0	0.0	0.0	10.5	11.1	18.5	24.0	45.8	15.8	23.5	12.9
MPNet [17]	31.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MTML [21]	40.2	-	14.5	54.0	79.2	48.8	42.7	32.4	32.7	21.9	10.9	0.8	14.2	39.9	42.1	64.3	96.5	36.4	70.8	21.5
3D-MPA [11]	59.1	35.3	51.9	72.2	83.8	**66.8**	63.0	**43.0**	44.5	58.4	38.8	31.1	43.2	**47.7**	**61.4**	**80.6**	**99.2**	50.6	**87.1**	40.3
PointGroup [20]	56.9	34.8	48.1	69.6	87.7	71.5	62.9	42.0	46.2	54.9	37.7	22.4	41.6	44.9	37.2	64.4	98.3	61.1	80.5	53.0
Ours-S	57.6	35.4	50.6	**73.8**	84.4	72.1	**69.9**	40.8	44.5	**62.4**	34.8	21.2	42.2	37.0	41.6	62.7	92.9	61.6	82.6	47.5
Ours-L	**61.0**	**40.6**	**52.3**	70.4	**90.2**	65.8	69.6	40.5	**47.2**	48.4	**44.7**	**34.9**	**52.3**	47.5	51.5	70.3	94.8	**74.3**	77.4	**56.4**

Table 4 – Per class 3D instance segmentation on ScanNetV2 [7] validation set. Both mAP and AP@50 are reported.

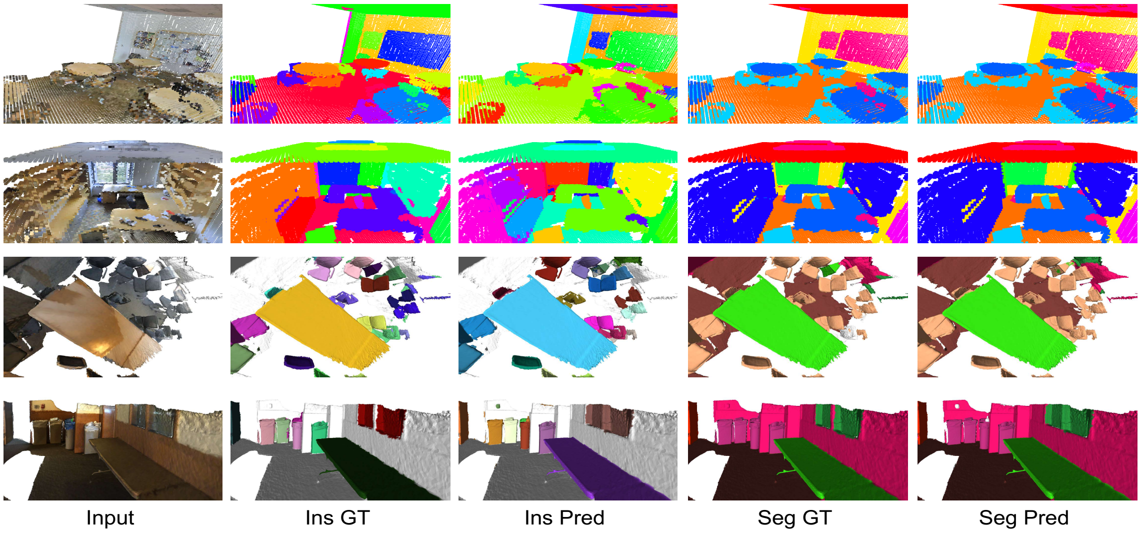


Figure 6 – Visualization of semantic and instance segmentation results on both S3DIS (top) and ScanNetv2 (bottom) benchmarks. Instances are presented with random colors.

Method	**AP@50**	bathtub	bed	bookshe.	cabinet	chair	counter	curtain	desk	door	otherfu.	picture	refrige.	s. curtain	sink	sofa	table	toilet	window
SGPN [38]	0.143	0.208	0.390	0.169	0.065	0.275	0.029	0.069	0.000	0.087	0.043	0.014	0.027	0.000	0.112	0.351	0.168	0.438	0.138
3D-BEVIS [10]	0.248	0.667	0.566	0.076	0.035	0.394	0.027	0.035	0.098	0.099	0.030	0.025	0.098	0.375	0.126	0.604	0.181	0.854	0.171
R-PointNet [44]	0.306	0.500	0.405	0.311	0.348	0.589	0.054	0.068	0.126	0.283	0.290	0.028	0.219	0.214	0.331	0.396	0.275	0.821	0.245
DPC [12]	0.355	0.500	0.517	0.467	0.228	0.422	0.133	0.405	0.111	0.205	0.241	0.075	0.233	0.306	0.445	0.439	0.457	0.974	0.23
3D-SIS [19]	0.382	1.000	0.432	0.245	0.190	0.577	0.013	0.263	0.033	0.320	0.240	0.075	0.422	0.857	0.117	0.699	0.271	0.883	0.235
MASC [22]	0.447	0.528	0.555	0.381	0.382	0.633	0.002	0.509	0.260	0.361	0.432	0.327	0.451	0.571	0.367	0.639	0.386	0.980	0.276
PanopticFusion [24]	0.478	0.667	0.712	0.595	0.259	0.550	0.000	0.613	0.175	0.250	0.434	0.437	0.411	0.857	0.485	0.591	0.267	0.944	0.35
3D-BoNet [42]	0.488	1.000	0.672	0.590	0.301	0.484	0.098	0.620	0.306	0.341	0.259	0.125	0.434	0.796	0.402	0.499	0.513	0.909	0.439
MTML [21]	0.549	1.000	0.807	0.588	0.327	0.647	0.004	0.815	0.180	0.418	0.364	0.182	0.445	1.000	0.442	0.688	0.571	1.000	0.396
PointGroup [20]	0.636	1.000	0.765	0.624	0.505	0.797	0.116	0.696	0.384	0.441	0.559	0.476	0.596	1.000	0.666	0.756	0.556	0.997	0.513
3D-MPA [11]	0.611	1.000	0.833	0.765	0.526	0.756	0.136	0.588	0.470	0.438	0.432	0.358	0.650	0.857	0.429	0.765	0.557	1.000	0.430
Ours	0.641	1.000	0.841	0.893	0.531	0.802	0.115	0.588	0.448	0.438	0.537	0.430	0.550	0.857	0.534	0.764	0.657	0.987	0.568

Table 5 – 3D instance segmentation results on ScanNetV2 testing set with AP@50 scores on 18 categories.

convolution offers a mechanism by which to have the segmentation method actively respond to the characteristics of the data at test time, and that this does in-fact improve robustness. It also allows devising an approach that avoids many other pitfalls associated with bottom-up methods. The particular dynamic-convolution-based method that we have proposed, DyCo3D, not only achieves state-of-the-art results, it offers improved efficiency over existing methods.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d seman-

tic parsing of large-scale indoor spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1, 3, 5

[2] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[3] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, and Chen Change Loy andDahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1

[5] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[8] Angela Dai and Matthias Nießner. 3DMV: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Eur. Conf. Comput. Vis.*, 2018. 2

[9] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Adv. Neural Inform. Process. Syst.*, 2016. 3

[10] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Bird's-eye-view instance segmentation. *arXiv preprint arXiv:1904.02199*, 2019. 8

[11] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi proposal aggregation for 3d semantic instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 2, 3, 5, 7, 8

[12] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *Int. Conf. Robotics & Automation*, 2020. 8

[13] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 3

[14] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 3

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Int. Conf. Comput. Vis.*, 2017. 1, 7, 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[17] Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In *Eur. Conf. Comput. Vis.*, 2020. 2, 3, 5, 6, 7, 8

[18] Tong He, Yifan Liu, Chunhua Shen, Xinlong Wang, and Changming Sun. Instance-aware embedding for point cloud instance segmentation. In *Eur. Conf. Comput. Vis.*, 2020. 2, 3, 5, 6, 7

[19] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3d semantic instance segmentation of rgb-d scans. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2, 7, 8

[20] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[21] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3d instance segmentation via multi-task metric learning. In *Int. Conf. Comput. Vis.*, 2019. 8

[22] Chen Liu and Yasutaka Furukawa. MASC: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019. 8

[23] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proc. IEEE Int. Conf. Intelligent Robots Syst.*, 2015. 2

[24] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *Proc. IEEE Int. Conf. Intelligent Robots Syst.*, 2019. 8

[25] Quang-Hieu Pham, Duc Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. JSIS3D: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 3

[26] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Int. Conf. Comput. Vis.*, 2019. 2, 4, 7

[27] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Graph attention convolution for point cloud semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[28] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 7

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[30] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[31] Charles R Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, 2017. 2

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015. 2

[33] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 7

[34] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Int. Conf. Comput. Vis.*, 2015. 2

[36] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Eur. Conf. Comput. Vis.*, 2020. 1, 3, 4, 5, 6

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017. 2, 3

[38] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity group proposal network for 3d point cloud instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 5, 6, 7, 8

[39] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 5, 6, 7

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2

[41] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Eur. Conf. Comput. Vis.*, 2018. 2

[42] 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. In *Adv. Neural Inform. Process. Syst.*, 2019. 2, 7, 8

[43] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-dense 3d object detector for point cloud. In *Int. Conf. Comput. Vis.*, 2019. 2

[44] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: Generative shape proposal network for 3d instance segmentation in point cloud. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 7, 8

[45] Lin Zhao and Wenbing Tao. JSNet: Joint instance and semantic segmentation of 3d point clouds. In *AAAI*, 2020. 3