3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation

Francis Engelmann[1,2†] Martin Bokeloh[2] Alireza Fathi[2] Bastian Leibe[1] Matthias Nießner[3]
[1]RWTH Aachen University [2]Google [3]Technical University Munich

Input: 3D Point Cloud *Object Center Votes & Aggregated Proposals* *Output: 3D Semantic Instances*

Figure 1: Given an input 3D point cloud, our Multi Proposal Aggregation network (3D-MPA) predicts point-accurate 3D semantic instances. We propose an object-centric approach which generates instance proposals followed by a graph convolutional network which enables higher-level interactions between adjacent proposals. Unlike previous methods, the final object instances are obtained by aggregating multiple proposals instead of pruning proposals using non-maximum-suppression.

Abstract

We present 3D-MPA, a method for instance segmentation on 3D point clouds. Given an input point cloud, we propose an object-centric approach where each point votes for its object center. We sample object proposals from the predicted object centers. Then, we learn proposal features from grouped point features that voted for the same object center. A graph convolutional network introduces inter-proposal relations, providing higher-level feature learning in addition to the lower-level point features. Each proposal comprises a semantic label, a set of associated points over which we define a foreground-background mask, an objectness score and aggregation features. Previous works usually perform non-maximum-suppression (NMS) over proposals to obtain the final object detections or semantic instances. However, NMS can discard potentially correct predictions. Instead, our approach keeps all proposals and groups them together based on the learned aggregation features. We show that grouping proposals improves over NMS and outperforms previous state-of-the-art methods on the tasks of 3D object detection and semantic instance segmentation on the ScanNetV2 benchmark and the S3DIS dataset.

† Work performed during internship at Google.

1. Introduction

With the availability of commodity RGB-D sensors such as Kinect or Intel RealSense, the computer vision and graphics communities have achieved impressive results on 3D reconstruction methods [27, 28] that can now even achieve global pose tracking in real time [8, 47]. In addition to the reconstruction of the geometry, semantic scene understanding is critical to many real-world computer vision applications, including robotics, upcoming applications on mobile devices, or AR/VR headsets. In order to understand reconstructed 3D environments, researchers have already made significant progress with 3D deep learning methods that operate on volumetric grids [6, 32, 37, 38, 48], point clouds [11, 31, 33], meshes [16, 36] or multi-view hybrids [7, 39]. While early 3D learning approaches focus mostly on semantic segmentation, we have recently seen many works on 3D semantic instance segmentation [18, 19, 49] and 3D object detection [29, 51], both of which we believe are critical for real-world 3D perception.

One of the fundamental challenges in 3D object detection lies in how to predict and process object proposals: On one side, top-down methods first predict a large number of rough object bounding box proposals (e.g., anchor mechanisms in Faster R-CNN [35]), followed by a second stage refinement step. Here, results can be generated in a single

forward pass, but there is little outlier tolerance to wrongly detected box anchors. On the other side, bottom-up approaches utilize metric-learning methods with the goal of learning a per-point feature embedding space which is subsequently clustered into object instances [10, 19, 24]. This strategy can effectively handle outliers, but it heavily depends on manually tuning cluster parameters and is inherently expensive to compute at inference time due to $O(N^2)$ pairwise relationships.

In this work, we propose 3D-MPA which follows a hybrid approach that takes advantage of the benefits of both top-down and bottom-up techniques: from an input point cloud representing a 3D scan, we generate votes from each point for object centers and group those into object proposals; then – instead of rejecting proposals using non-maximum-suppression – we learn higher-level features for each proposal, which we use to cluster the proposals into final object detections. The key idea behind this strategy is that the number of generated proposals is orders of magnitude smaller than the number of raw input points in a 3D scan, which makes grouping computationally very efficient. At the same time, each object can receive multiple proposals, which simplifies proposal generation since objects of all sizes are handled in the same fashion, and we can easily tolerate outlier proposals further down the pipeline.

To this end, our method first generates object-centric proposals using a per-point voting scheme from a sparse volumetric feature backbone. We then interpret the proposals as nodes of a proposal graph which we feed into a graph convolutional neural network in order to enable higher-order interactions between neighboring proposal features. In addition to proposal losses, the network is trained with a proxy loss between proposals similar to affinity scores in metric learning; however, due to the relatively small number of proposals, we can efficiently train the network and cluster proposals. In the end, each node predicts a semantic class, an object foreground mask, an objectness score, and additional features that are used to group nodes together.

In summary, our contributions are the following:

- A new method for 3D instance segmentation based on dense object center prediction leveraging learned semantic features from a sparse volumetric backbone.
- To obtain the final object detections and semantic instances from the object proposals, we replace the commonly used NMS with our multi proposal aggregation strategy based on jointly learned proposal features and report significantly improved scores over NMS.
- We employ a graph convolutional network that explicitly models higher-order interactions between neighboring proposal features in addition to the lower-level point features.

2. Related Work

Object Detection and Instance Segmentation. In the 2D domain, object detection and instance segmentation have most notably been influenced by Faster R-CNN from Ren *et al*. [35], which introduced the anchor mechanism to predict proposals with associated objectness scores and regions of interest that enable the regression of semantic bounding boxes. This approach was extended in Mask-RCNN [17] to predict per-pixel object instance masks. Hou *et al*. [18] apply the 2D proposal ideas onto the 3D domain by means of dense 3D convolutional networks. As an alternative, proposal-free methods were proposed in [4, 14, 19] which rely on metric learning. In the 2D domain, Fathi *et al*. [14] estimate how likely pixels are to belong to the same object. De Brabandere *et al*. [4] define a discriminative loss, which moves feature points of the same object towards their mean while pushing means of different objects apart. This discriminative loss is adopted by Lahoud *et al*. [19] to perform instance segmentation in 3D space. Final instances are obtained via clustering of the learned feature space. Yang *et al*. [49] directly predict object bounding boxes from a learned global feature vector and obtain instance masks by segmenting points inside a bounding box. The recent VoteNet [29] highlights the challenge of directly predicting bounding box centers in sparse 3D data as most surface points are far away from object centers. Instead, they predict bounding boxes by grouping points from the same object based on their votes for object centers. We adopt the object-centric approach, extend it with a branch for instance mask prediction and replace NMS with a grouping mechanism of jointly-learned proposal features.

3D Deep Learning. PointNets [31] have pioneered the use of deep learning methods for point cloud processing. Since then, we have seen impressive progress in numerous different fields, including 3D semantic segmentation [15, 12, 21, 31, 33, 40, 46], 3D instance segmentation [10, 18, 19, 45, 49, 50], object detection [18, 29, 51] and relocalization [42], flow estimation [3, 25, 43], scene-graph reconstruction [1] and scene over-segmentation [20]. Point-based architectures, such as PointNet [29] and PointNet++ [34] operate directly on unstructured sets of points, while voxel based approaches, such as 3DMV [7] or SparseConvNets [5, 15] transform the continuous 3D space into a discrete grid representation and define convolutional operators on the volumetric grid, analogously to image convolutions in the 2D domain. Graph-based approaches [22, 41, 46] define convolutional operators over graph-structured data such as 3D meshes [16, 36], citation networks [41], or molecules [9]. Here, we leverage the voxel-based approach of Graham *et al*. [15] as point feature backbone and use the graph neural network of Wang *et al*. [46] to enable higher-level interactions between proposals.

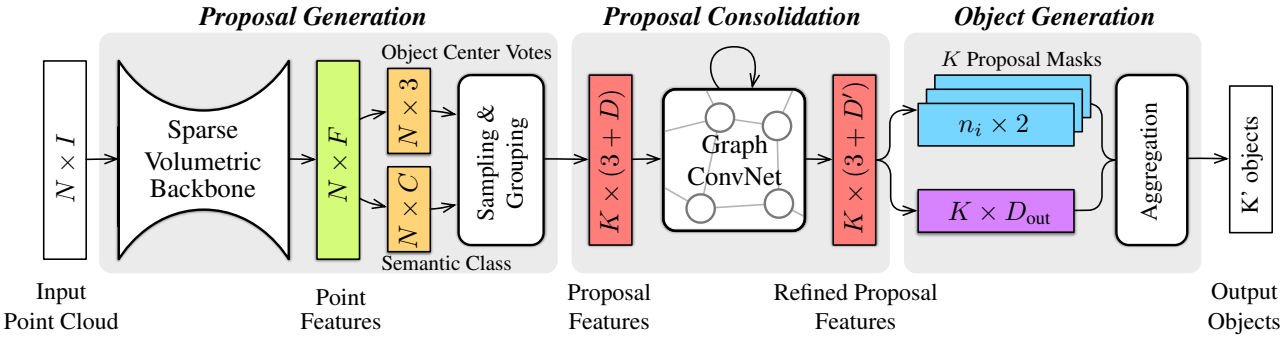

Figure 2: **3D-MPA network architecture**. From an input point cloud, our network predicts object instance masks by aggregating object proposal masks. The full model consists of three parts: the proposal generation *(left)* follows an object-centric strategy: each point votes for the center of the object it belongs to. Proposal positions are then sampled from the predicted object centers. By grouping and aggregating votes in the vicinity of sampled proposal positions, we learn proposal features. During proposal consolidation *(middle)*, proposal features are further refined using a graph convolutional network, which enables higher-order interactions on the level of proposals. Finally, we propose to aggregate multiple proposals by clustering jointly learned aggregation features as opposed to the commonly used non-maximum-suppression *(right)*.

3. Method

The overall architecture of 3D-MPA is depicted in Fig. 2. The model consists of three parts: the first one takes as input a 3D point cloud and learns object proposals from sampled and grouped point features that voted for the same object center (Sec. 3.1). The next part consolidates the proposal features using a graph convolutional network enabling higher-level interactions between proposals which results in refined proposal features (Sec. 3.2). Last, the object generator consumes the object proposals and generates the final object detections, *i.e.* semantic instances. We parameterize an object as a set of points associated with that object and a semantic class. (Sec. 3.3).

3.1. Proposal Generation

Given a point cloud of size $N \times I$, consisting of N points and I- dimensional input features (*e.g.* positions, colors and normals), the first part of the network generates a fixed number K of object proposals. A proposal is a tuple (y_i, g_i, s_i) consisting of a position $y_i \in \mathbb{R}^3$, a proposal features vector $g_i \in \mathbb{R}^D$ and a set of points s_i associated with the proposal.

To generate proposals, we need strong point features that encode the semantic context and the geometry of the underlying scene. We implement a sparse volumetric network [5, 15] as feature backbone to generate per-point features $\{f_i \in \mathbb{R}^F\}_{i=1}^N$ (Fig. 2, ■). Semantic context is encoded into the point features by supervising the feature backbone with semantic labels, using the standard cross-entropy loss for per-point semantic classification $\mathcal{L}_{\text{sem.pt.}}$. Following the object-centric approach suggested by Qi *et al.* [29], points vote for the center of the object they belong to. However, unlike [29], only points from objects predict a center. This is possible since we jointly predict semantic classes, *i.e.* we can differentiate between points from foreground (objects) and background (walls, floor, *etc.*) during both training and test. This results in precise center predictions since noisy predictions from background points are ignored. In particular, this is implemented as a regression loss which predicts per-point relative 3D offsets $\Delta x_i \in \mathbb{R}^3$ between a point position $x_i \in \mathbb{R}^3$ and its corresponding ground truth bounding-box center $c_i^* \in \mathbb{R}^3$. We define the per-point center regression loss as:

$$\mathcal{L}_{\text{cent.pt.}} = \frac{1}{M} ||x_i + \Delta x_i - c_i^*||_H \cdot \mathbb{1}(x_i) \,, \qquad (1)$$

where $||\cdot||_H$ is the *Huber*-loss (or smooth L_1-loss) and $\mathbb{1}(\cdot)$ is a binary function indicating whether a point x_i belongs to an object. M is a normalization factor equal to the total number of points on objects. All in all, the feature backbone has two heads (Fig. 2, ■): a semantic head (which performs semantic classification of points) and a center head (which regresses object centers for each point). They are jointly supervised using the combined loss $\mathcal{L}_{\text{point}}$ where λ is a weighting factor set to 0.1:

$$\mathcal{L}_{\text{point}} = \lambda \cdot \mathcal{L}_{\text{sem.pt.}} + \mathcal{L}_{\text{cent.pt.}}. \qquad (2)$$

Proposal Positions and Features. After each point (that belongs to an abject) has voted for a center, we obtain a distribution over object centers (Fig. 3, 3rd col.). From this distribution, we randomly pick K samples as proposal *positions* $\{y_i = x_i + \Delta x_i \in \mathbb{R}^3\}_{i=1}^K$ (Fig. 3, 4th col.). We found random sampling to work better than *Farthest Point Sampling* (FPS) used in [29], as FPS favors outliers far away from true object centers. Next, we define the set of associated points s_i as those points that voted for centers within a radius r of the sampled proposal position y_i. The proposal

features $\{g_i \in \mathbb{R}^D\}_{i=1}^K$ are learned using a PointNet [31] applied to the point features of the associated points s_i. This corresponds to the grouping and normalization technique described in [29]. At this stage, we have K proposals composed of 3D positions y_i located near object centers, proposal features $g_i \in \mathbb{R}^D$ describing the local geometry and the semantics of the nearest objects (Fig. 2, ■), along with a set of points s_i associated with each proposal.

3.2. Proposal Consolidation

So far, proposal features encode *local* information of their associated objects. During proposal consolidation, proposals become aware of their *global* neighborhood by explicitly modeling higher-order interactions between neighboring proposals. To this end, we define a *graph convolutional network* (GCN) over the proposals. While the initial point-feature backbone operates at the level of points, the GCN operates at the level of proposals. In particular, the nodes of the graph are defined by the proposal positions y_i with associated proposal features g_i. An edge between two nodes exists if the Euclidean distance d between two 3D proposal positions $y_{\{i,j\}}$ is below 2 m. We adopt the convolutional operator from DGCNN [46] to define edge-features e_{ij} between two neighboring proposals as:

$$e_{ij} = h_\Theta\big([y_i, g_i], [y_j, g_j] - [y_i, g_i]\big) \;, \tag{3}$$

where h_Θ is a non-linear function with learnable parameters θ and $[\cdot,\cdot]$ denotes concatenation. The graph convolutional network consists of l stacked graph convolutional layers. While our method also works without the GCN refinement (*i.e.* $l = 0$), we observe the best results using $l = 10$ (Sec. 4). To conclude, during proposal consolidation a GCN learns refined proposal features $\{h_i \in \mathbb{R}^{D'}\}_{i=1}^K$ given the initial proposal features $\{g_i \in \mathbb{R}^D\}_{i=1}^K$ (Fig. 2, ■).

3.3. Object Generation

At this stage, we have K proposals $\{(y_i, h_i, s_i)\}_{i=1}^K$ with positions y_i, refined features h_i and sets of points s_i. The goal is to obtain the final semantic instances (or object detections) from these proposals. To this end, we predict for every proposal a semantic class, an aggregation feature vector, an objectness score and a binary foreground-background mask over the points s_i associated with the proposal. Specifically, the proposal features h_i are input to an MLP with output sizes $(128, 128, D_{out})$ where $D_{out} = S + E + 2$ with S semantic classes, E-dimensional aggregation feature and a 2D (positive, negative) objectness score (Fig. 2, ■).

The objectness score [29, 35] classifies proposals into positive or negative examples. It is supervised via a cross-entropy loss $\mathcal{L}_{obj.}$. Proposals near a ground truth center (< 0.3 m) are classified as positive. They are classified as negative, if they are far away (> 0.6 m) from any ground truth center, or if they are equally far away from two ground truth centers since then the correct ground truth object is ambiguous. This is the case when $d_1 > 0.6 \cdot d_2$ where d_i is the distance to the i^{th} closest ground truth center.

Positive proposals are further supervised to predict a semantic class, aggregation features, and a binary mask. Negative ones are ignored. We use a cross-entropy loss $\mathcal{L}_{\text{sem.}}$ to predict the semantic label of the closest ground truth object.

Aggregation Features. Previous methods such as VoteNet [29] or 3D-BoNet [49] rely on non-maximum-suppression (NMS) to obtain the final objects. NMS iteratively selects proposals with the highest objectness score and removes all others that overlap with a certain IoU. However, this is sensitive to the quality of the objectness scores and can discard correct predictions. Instead of rejecting potentially useful information, we combine multiple proposals. To this end, we learn aggregation features for each proposal which are then clustered using DBScan [13].

All proposals whose aggregation features end up in the same cluster are aggregated together, yielding the final object detections. The points of a final object are the union over the foreground masks of combined proposals. As the number of proposals is relatively small ($K \approx 500$) compared to the full point cloud ($N \approx 10^6$), this step is very fast (~ 8 ms). This is a significant advantage over clustering full point clouds [10, 19], which can be prohibitively slow.

We investigate two types of aggregation features:
① *Geometric features* $\{\epsilon_i \in \mathbb{R}^{E=4}\}_{i=1}^K$ are composed of a refined 3D object center prediction Δy_i and a 1D object radius estimation r_i. The loss is defined as:

$$\mathcal{L}_{\text{agg.}} = ||y_i + \Delta y_i - c_i^*||_H + ||r_i - r_i^*||_H \tag{4}$$

where c_i^* is the nearest ground truth object center and r_i^* the radius of the nearest ground truth object bounding sphere.
② *Embedding features* $\{\epsilon_i \in \mathbb{R}^E\}_{i=1}^K$ are supervised with a discriminative loss function [4]. This loss was already successfully applied for 3D instance segmentation [10, 19]. It is composed of three terms: $\mathcal{L}_{\text{agg.}} = \mathcal{L}_{\text{var.}} + \mathcal{L}_{\text{dist.}} + \gamma \cdot \mathcal{L}_{\text{reg.}}$

$$\mathcal{L}_{\text{var.}} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_C}\sum_{i=1}^{N_C}[||\mu_C - \epsilon_i|| - \delta_v]_+^2 \tag{5}$$

$$\mathcal{L}_{\text{dist.}} = \frac{1}{C(C-1)}\sum_{\substack{C_A=1\\C_A\neq C_B}}^{C}\sum_{C_B=1}^{C}[2\delta_d - ||\mu_{C_A} - \mu_{C_B}||]_+^2 \tag{6}$$

$$\mathcal{L}_{\text{reg.}} = \frac{1}{C}\sum_{C=1}^{C}||\mu_C|| \tag{7}$$

In our experiments, we set $\gamma = 0.001$ and $\delta_v = \delta_d = 0.1$. C is the total number of ground truth objects and N_C the number of proposals belonging to one object. $\mathcal{L}_{\text{var.}}$ pulls features that belong to the same instance towards their mean, $\mathcal{L}_{\text{dist.}}$

pushes clusters with different instance labels apart, and $\mathcal{L}_{\text{reg.}}$ is a regularization term pulling the means towards the origin. Further details and intuitions are available in the original work by DeBrabandere *et al*. [4]. In Sec. 4, we will show that geometric features outperform embedding features.

Mask Prediction. Each positive proposal predicts a class-agnostic binary segmentation mask over the points s_i associated with that proposal, where the number of points per proposal i is $|s_i| = n_i$ (Fig. 2, ■). Prior approaches obtain masks by segmenting 2D *regions of interest* (RoI) (Mask-RCNN [17]) or 3D bounding boxes (3D-BoNet [49]). Since we adopt an object-centric approach, mask segmentation can directly be performed on the points s_i associated with a proposal. In particular, for each proposal, we select the per-point features f_i of points that voted for a center within a distance r of the proposal position y_i. Formally, the set of selected per-point features is defined as $M_f = \{f_i \,|\, \|(x_i + \Delta x_i) - y_i\|_2 < r\}$ with $r = 0.3\,\text{m}$. The selected features M_f are passed to a PointNet [32] for binary segmentation, *i.e*., we apply a shared MLP on each per-point feature, compute max-pooling over all feature channels, and concatenate the result to each feature before passing it through another MLP with feature sizes (256, 128, 64, 32, 2). Points that have the same ground truth instance label as the closest ground truth object instance label are supervised as foreground, while all others are background. Similar to [49], the mask loss $\mathcal{L}_{\text{mask}}$ is implemented as *FocalLoss* [23] instead of a cross-entropy loss to cope with the foreground-background class imbalance.

3.4. Training Details

The model is trained end-to-end from scratch using the multi-task loss $\mathcal{L} = \mathcal{L}_{\text{point}} + \mathcal{L}_{\text{obj.}} + 0.1 \cdot \mathcal{L}_{\text{sem.}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{agg.}}$. The batch size is 4 and the initial learning rate 0.1 which is reduced by half every $2{\cdot}10^4$ iterations and trained for $15{\cdot}10^4$ iterations in total. Our model is implemented in TensorFlow and runs on an Nvidia TitanXp GPU (12GB).

Input and data augmentation. Our network is trained on $3\,\text{m} \times 3\,\text{m}$ point cloud crops of N points sampled from the surface of a 3D mesh. During test time, we evaluate on full scenes. Input features are the 3D position, color and normal assigned to each point. Data augmentation is performed by randomly rotating the scene by Uniform$[-180°, 180°]$ around the upright axis and Uniform$[-10°, 10°]$ around the other axis. The scenes are randomly flipped in both horizontal directions and randomly scaled by Uniform$[0.9, 1.1]$.

4. Experiments

We compare our approach to previous state-of-the-art methods on two large-scale 3D indoor datasets (Sec. 4.1). Our ablation study analyzes the contribution of each component of our model and shows in particular the improvement of aggregating proposals over NMS (Sec. 4.2).

3D Object Detection		
ScanNetV2	mAP@25%	mAP@50%
DSS [37]	15.2	6.8
MRCNN 2D-3D [17]	17.3	10.5
F-PointNet [30]	19.8	10.8
GSPN [50]	30.6	17.7
3D-SIS [18]	40.2	22.5
VoteNet [29]	58.6	33.5
3D-MPA (Ours)	**64.2**	**49.2**

Table 1: **3D object detection scores on ScanNetV2** [6] validation set. We report per-class mean average precision (mAP) with an IoU of 25 % and 50 %. The IoU is computed on bounding boxes. All other scores are as reported in [29].

3D Instance Segmentation		
S3DIS 6-fold CV	mAP@50%	mAR@50%
PartNet [26]	56.4	43.4
ASIS [45]	63.6	47.5
3D-BoNet [49]	65.6	47.6
3D-MPA (Ours)	**66.7**	**64.1**
S3DIS Area 5	mAP@50%	mAR@50%
ASIS [45]	55.3	42.4
3D-BoNet [49]	57.5	40.2
3D-MPA (Ours)	**63.1**	**58.0**

Table 2: **3D instance segmentation scores on S3DIS [2].** We report scores on Area 5 *(bottom)* and 6-fold cross validation results *(top)*. The metric is mean average precision (mAP) and mean average recall (mAR) at an IoU threshold of 50%. The IoU is computed on per-point instance masks.

3D Instance Segmentation						
ScanNetV2	Validation Set			Hidden Test Set		
	mAP	@50%	@25%	mAP	@50%	@25%
SGPN [44]	-	11.3	22.2	4.9	14.3	39.0
3D-BEVIS [10]	-	-	-	11.7	24.8	40.1
3D-SIS [18]	-	18.7	35.7	16.1	38.2	55.8
GSPN [50]	19.3	37.8	53.4	15.8	30.6	54.4
3D-BoNet [49]	-	-	-	25.3	48.8	68.7
MTML [19]	20.3	40.2	55.4	28.2	54.9	73.1
3D-MPA (Ours)	**35.3**	**59.1**	**72.4**	**35.5**	**61.1**	**73.7**

Table 3: **3D instance segmentation scores ScanNetV2** [6]. The metric is mean average precision (mAP) at an IoU threshold of 55%, 50% and averaged over the range [0.5:0.95:05]. IoU on per-point instance masks.

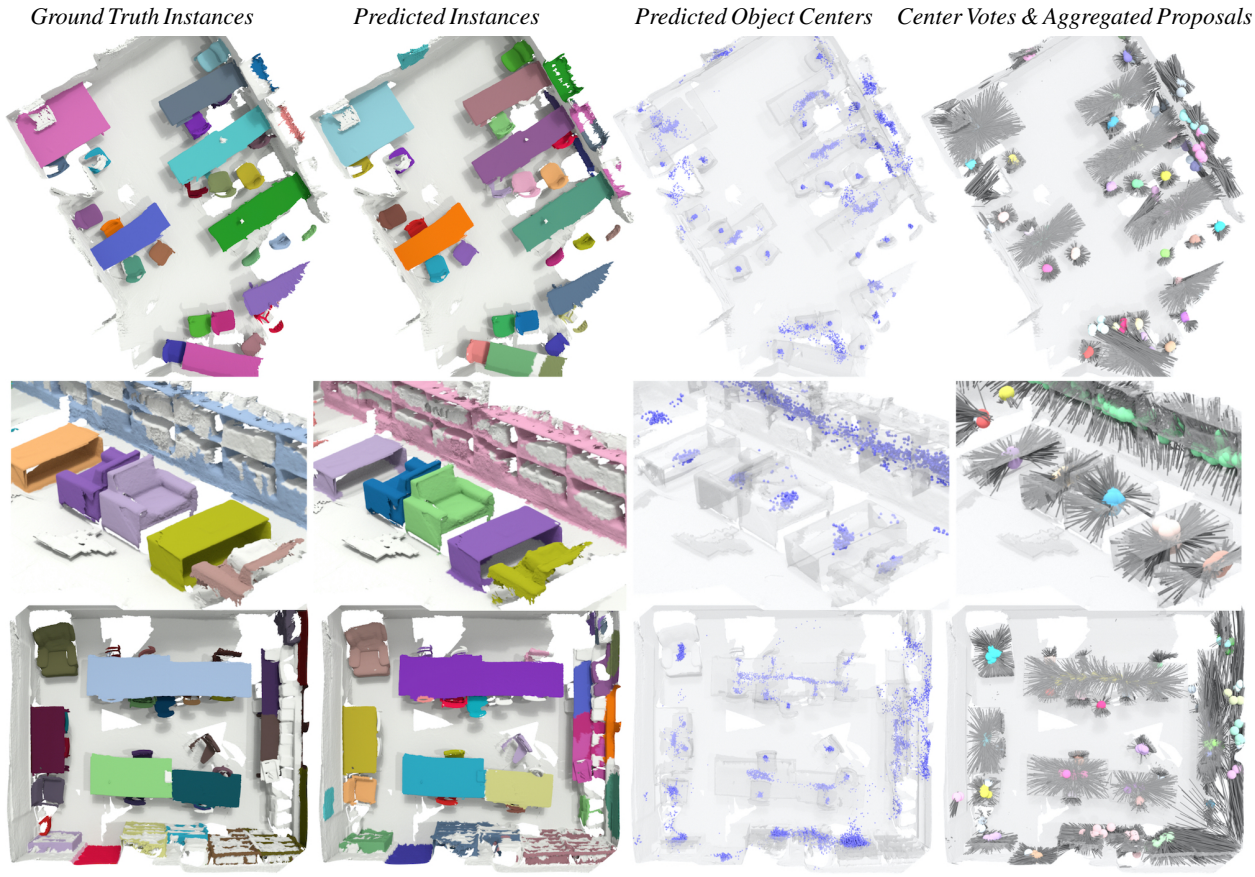

Figure 3: **Qualitative results and intermediate steps** on ScanNetV2 [6]. First two columns: Our approach properly segments instances of vastly different sizes and makes clear decisions at object boundaries. Different colors represent separate instances (ground truth and predicted instances are not necessarily the same color). Third column: Every point on the surface of an object predicts its object center. These centers are shown as blue dots. Fourth column: Gray segments correspond to votes, they illustrate which point predicted a center. Colored spheres represent proposals. Proposals are obtained by sampling from the predicted object centers. Proposal features are learning from grouped point features that voted for the same object center. Spheres with the same color show which proposals are grouped together based on these learned proposal features.

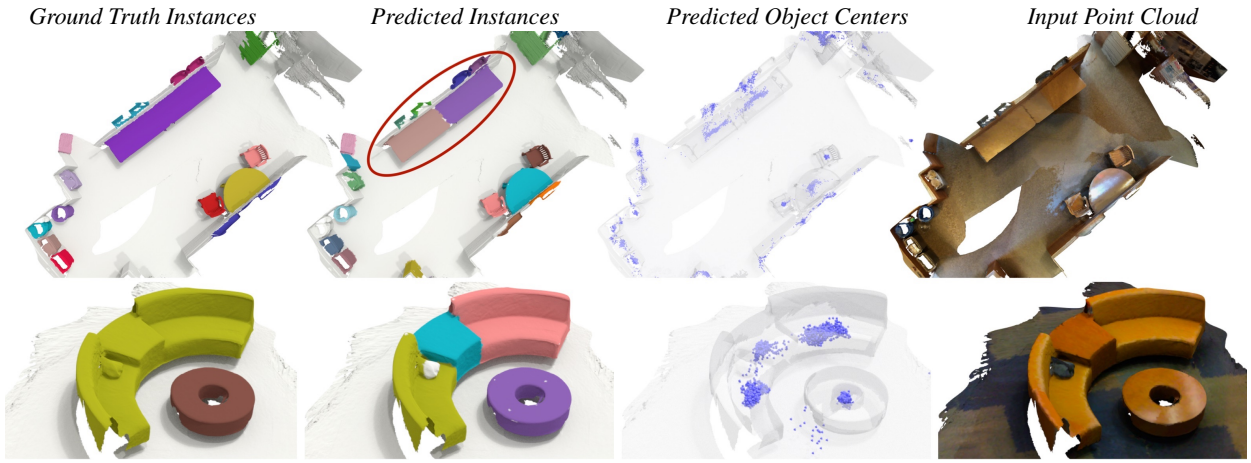

Figure 4: **Failure Cases.** We show two failure cases where our method incorrectly separates single instances. However, when comparing them to the input point cloud, they are still plausible predictions.

mAP@25 %	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
SegCluster [18]	11.8	13.5	18.9	14.6	13.8	11.1	11.5	11.7	0.0	13.7	12.2	12.4	11.2	18.0	19.5	18.9	16.4	12.2	13.4
MRCNN [17]	15.7	15.4	16.4	16.2	14.9	12.5	11.6	11.8	19.5	13.7	14.4	14.7	21.6	18.5	25.0	24.5	24.5	16.9	17.1
SGPN [44]	20.7	31.5	31.6	40.6	31.9	16.6	15.3	13.6	0.0	17.4	14.1	22.2	0.0	0.0	72.9	52.4	0.0	18.6	22.2
3D-SIS [18]	32.0	66.3	65.3	56.4	29.4	26.7	10.1	16.9	0.0	22.1	35.1	22.6	28.6	37.2	74.9	39.6	57.6	21.1	35.7
MTML [19]	34.6	80.6	**87.7**	**80.3**	67.4	45.8	47.2	45.3	19.8	9.7	49.9	54.2	44.1	74.9	98.0	44.5	79.4	33.5	55.4
3D-MPA (Ours)	**69.9**	**83.4**	87.6	76.1	**74.8**	**56.6**	**62.2**	**78.3**	**48.0**	**62.5**	**69.2**	**66.0**	**61.4**	**93.1**	**99.2**	**75.2**	**90.3**	**48.6**	**72.4**

Table 4: **Per class 3D instance segmentation** on ScanNetV2 [6] validation set with mAP@25% on 18 classes. Our method outperforms all other methods on all classes except for *chair* and *sofa*.

mAP@50%	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
SegCluster [18]	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	10.8
MRCNN [17]	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	9.1
SGPN [44]	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	11.3
3D-SIS [18]	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	18.7
MTML [19]	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	40.2
3D-MPA (Ours)	**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**	**59.1**

Table 5: **Per class 3D instance segmentation** on ScanNetV2 [6] validation set with mAP@50% on 18 classes. Our method outperforms all other methods on all classes.

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

Datasets. The *ScanNetV2* [6] benchmark dataset consists of richly-annotated 3D reconstructions of indoor scenes. It comprises 1201 training scenes, 312 validation scenes and 100 hidden test scenes. The benchmark is evaluated on 20 semantic classes which include 18 different object classes.

The *S3DIS* [2] dataset is a collection of six large-scale indoor areas annotated with 13 semantic classes and object instance labels. We follow the standard evaluation protocol and report scores on Area 5, as well as 6-fold cross validation results over all six areas.

Object detection scores are shown in Tab. 1. Object detections are obtained by fitting a tight axis-aligned bounding box around the predicted object point-masks. We compare 3D-MPA to recent approaches including *VoteNet* [29] on the ScanNetV2 [6] dataset. Scores are obtained by using the evaluation methodology provided by [29]. Our method outperforms all previous methods by at least **+5.8 mAP@25%** and **+15.7 mAP@50%**.

Instance segmentation scores on S3DIS [2] are shown in Tab. 2. Per-class instance segmentation results are shown in Tab. 7. We report mean average precision (mAP) and mean average recall (mAR) scores. Our scores are computed using the evaluation scripts provided by Yang *et al.* [49]. Our approach outperforms all previous methods. In particular, we report an increased recall of **+17.8 mAR@50%** on Area5 and **+16.5 mAR@50%** on 6-fold cross validation, which means we detect significantly more objects, while simultaneously achieving higher precision.

We show results on ScanNetV2 [6] validation and hidden test set in Tab. 3 and per-class scores with mAP@25% in Tab. 4 and mAP@50% in Tab. 5. We improve over previous methods by at least **+18.1 mAP@50%** and **+17.0 mAP@25%**. In particular, our 3D-MPA outperforms all other methods in every object class on mAP@50 (Tab. 5). On mAP@25, we outperform on all classes except *chair* and *sofa*. Qualitative results on ScanNetV2 are visualized in Fig. 3 and failure cases in Fig. 4.

4.2. Ablation study

In Tab. 6, we show the result of our ablation study analyzing the design choices of each component of our model. The evaluation metric is mean average precision (mAP) on the task of instance segmentation, evaluated on the ScanNetV2 validation set.

Ablation Study	
3D Instance Segmentation (ScanNetV2 val.)	mAP@50%
① Proposals + NMS	47.5
② Agg. Props. (proposal positions)	52.4 (+4.9)
③ Agg. Props. (embedding features)	56.7 (+9.2)
④ Agg. Props. (geometric features)	57.8 (+10.3)
⑤ Agg. Props. (geometric features + GCN)	**59.1** (+11.6)

Table 6: **Ablation study.** In Sec. 4.2 we discuss the results in detail. Scores are instance segmentation results on the ScanNetV2 [6] validation set and absolute improvements in mAP (in green) relative to the baseline ①.

	S3DIS 6-fold CV	ceil.	floor	walls	beam	colm,	wind.	door	table	chair	sofa	bookc.	board	clut.	**mean**
mAP@0.5	3D-BoNet [49]	88.5	89.9	**64.9**	42.3	48.0	**93.0**	66.8	**55.4**	72.0	49.7	**58.3**	**80.7**	**47.6**	65.6
	3D-MPA (Ours)	**95.5**	**99.5**	59.0	**44.6**	**57.7**	89.0	**78.7**	34.5	**83.6**	**55.9**	51.6	71.0	46.3	**66.7**
mAR@0.5	3D-BoNet [49]	61.8	74.6	50.0	42.2	27.2	62.4	58.5	**48.6**	64.9	28.8	28.4	46.5	28.6	46.7
	3D-MPA (Ours)	**68.4**	**96.2**	**51.9**	**58.8**	**77.6**	**79.8**	**69.5**	32.8	**75.2**	**71.1**	**46.2**	**68.2**	**38.2**	**64.1**

Table 7: **Per class 3D instance segmentation scores on S3DIS** [2]. We report per-class mean average precision (mAP) and recall (mAR) with an IoU of 50 %. 3D-BoNet are up-to-date numbers provided by the original authors. Our method detects significantly more objects (+ 17.4 mAR) and it is even able to do so with a higher precision (+ 1.1 mAP).

Effect of grouping compared to NMS. The main result of this work is that grouping multiple proposals is superior to non-maximum-suppression (NMS). We demonstrate this experimentally by comparing two baseline variants of our model: In experiment ① (Tab. 6), we apply the traditional approach of predicting a number of proposals and applying NMS to obtain the final predictions. The model corresponds to the one depicted in Fig. 2 without proposal consolidation and with the aggregation replaced by NMS. NMS chooses the most confident prediction and suppresses all other predictions with an IoU larger than a specified threshold, in our case 25%. For experiment ②, we perform a naive grouping of proposals by clustering the proposal positions y_i. The final object instance masks are obtained as the union over all proposal masks in one cluster. We observe a significant increase of **+4.9 mAP** by replacing NMS with aggregation.

How important are good aggregation features? In experiment ②, we group proposals based on their position y_i. These are still relatively simple features. In experiments ③ and ④, proposals are grouped based on learned embedding features and learned geometric features, respectively. These features are described in Sec. 3.3. Again, we observe a notable improvement of +5.4 mAP compared to experiment ② and even **+10.3 mAP** over ①. In our experiments, the geometric features performed better than the embedding features (+1.1 mAP). One possible explanation could be that the geometric features have an explicit meaning and are therefore easier to train than the 5-dimensional embedding space used in experiment ③. Therefore, for the next experiment in the ablation study and our final model, we make use of the geometric features. In summary, the quality of the aggregation features has a significant impact.

Does the graph convolutional network help? The graph convolutional network (GCN) defined on top of proposals enables higher-order interaction between proposals. Experiment ⑤ corresponds to the model depicted in Fig. 2 with a 10 layer GCN. Experiment ④ differs from experiment ⑤ in that it does not include the GCN for proposal consolidation. Adding the GCN results in another improvement of +1.3 mAP. In total, by incorporating the GCN and replacing NMS with multi-proposal aggregation, we observe an improvement of **+11.6 mAP** over the same network architecture without those changes.

5. Conclusion

In this work, we introduced 3D-MPA, a new method for 3D semantic instance segmentation. Our core idea is to combine the benefits of both top-down and bottom-up object detection strategies. That is, we first produce a number of proposals using an object-centric voting scheme based on a sparse volumetric backbone. Each object may receive multiple proposals, which makes our method robust to potential outliers in the object proposal stage. However, at the same time we obtain only a handful of proposals such that clustering them is computationally inexpensive. To address this, we first allow higher-order feature interactions between proposals via a graph convolutional network. We then aggregate proposals based on graph relationship results and proposal feature similarities. We show that graph convolutions help to achieve high evaluation scores, although, the largest improvement originates from our multi proposal aggregation strategy. Our combined approach achieves state-of-the-art instance segmentation and object detection results on the popular ScanNetV2 and S3DIS datasets, thus validating our algorithm design.

Overall, we believe that multi proposal aggregation is a promising direction for object detection, in particular in the 3D domain. However, there still remain many interesting future avenues, for instance, how to combine detection with tracking in semi-dynamic sequences. We see a variety of interesting ideas, where proposals could be distributed in 4D space and accumulated along the time-space axis.

Acknowledgements. We would like to thank Theodora Kontogianni, Jonas Schult, Jonathon Luiten, Mats Steinweg, Ali Athar, Dan Jia and Sabarinath Mahadevan for helpful feedback as well as Angela Dai for help with the video. This work was funded by the ERC Consolidator Grant DeeViSe (ERC-2017-COG-773161) and the ERC Starting Grant Scan2CAD (804724).

References

[1] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[2] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 7, 8

[3] A. Behl, D. Paschalidou, S. Donne, and A. Geiger. PointFlowNet: Learning Representations for Rigid Motion Estimation from Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[4] B. D. Brabandere, D. Neven, and L. V. Gool. Semantic Instance Segmentation with a Discriminative Loss Function. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR'W)*, 2017. 2, 4, 5

[5] C. Choy, J. Gwak, and S. Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 5, 6, 7

[7] A. Dai and M. Nießner. 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[8] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time Globally Consistent 3D Reconstruction Using On-the-fly Surface Reintegration. *ACM Transactions on Graphics (TOG)*, 2017. 1

[9] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Neural Information Processing Systems (NIPS)*, 2015. 2

[10] C. Elich, F. Engelmann, J. Schult, T. Kontogianni, and B. Leibe. 3D-BEVIS: Birds-Eye-View Instance Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2019. 2, 4, 5

[11] F. Engelmann, T. Kontogianni, and B. Leibe. Dilated Point Convolutions: On the Receptive Field Size of Point Convolutions on 3D Point Clouds. In *International Conference on Robotics and Automation (ICRA)*, 2020. 1

[12] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *European Conference on Computer Vision Workshop (ECCV'W)*, 2018. 2

[13] M. Ester, H. peter Kriegel, J. Sander, and X. Xu. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases With Noise. In *ACM International Conference on Knowledge Discovery & Data Mining (KDD)*, 1996. 4

[14] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic Instance Segmentation via Deep Metric Learning. *CoRR*, abs/1703.10277, 2017. 2

[15] B. Graham, M. Engelcke, and L. van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[16] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. MeshCNN: A Network with an Edge. *ACM Transactions on Graphics (TOG)*, 2019. 1, 2

[17] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 7

[18] J. Hou, A. Dai, and M. Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 7

[19] J. Lahoud, B. Ghanem, M. Pollefeys, and M. R. Oswald. 3D Instance Segmentation via Multi-Task Metric Learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 4, 5, 7

[20] L. Landrieu and M. Boussaha. Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[21] L. Landrieu and M. Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[22] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated Graph Sequence Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[23] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[24] C. Liu and Y. Furukawa. MASC: Multi-scale Affinity with Sparse Convolution for 3D Instance Segmentation. *CoRR*, abs/1902.04478, 2017. 2

[25] X. Liu, C. R. Qi, and L. J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[26] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[27] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1

[28] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Transactions on Graphics (TOG)*, 2013. 1

[29] C. R. Qi, O. Litany, K. He, and L. J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 4, 5, 7

[30] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4

[32] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NIPS)*, 2017. 1, 2

[34] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3D Graph Neural Networks for RGBD Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1, 2, 4

[36] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe. DualConvMesh-Net: Joint Geodesic and Euclidean Convolutions on 3D Meshes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[37] S. Song and J. Xiao. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 5

[38] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C. J. Kuo. SPG-Net: Segmentation Prediction and Guidance Network for Image Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[39] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[40] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. Tangent Convolutions for Dense Prediction in 3D. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[41] K. Thomas and W. Max. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[42] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[43] S. Wang, S. Suo, W. Ma, A. Pokrovsky, and R. Urtasun. Deep Parametric Continuous Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[44] W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 7

[45] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia. Associatively Segmenting Instances and Semantics in Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5

[46] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. In *ACM Transactions on Graphics (TOG)*, 2019. 2, 4

[47] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison. ElasticFusion: Dense SLAM without a Pose Graph. In *Robotics: Science and Systems (RSS)*, 2015. 1

[48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[49] 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 *Neural Information Processing Systems (NIPS)*, 2019. 1, 2, 4, 5, 7, 8

[50] L. Yi, W. Zhao, H. Wang, M. Sung, and L. Guibas. GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5

[51] Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2