RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction

Yinyu Nie\textsuperscript{1,2} \hspace{1cm} Ji Hou\textsuperscript{3} \hspace{1cm} Xiaoguang Han\textsuperscript{1,*} \hspace{1cm} Matthias Nießner\textsuperscript{3}

\textsuperscript{1}SRIBD, CUHKSZ \hspace{1cm} \textsuperscript{2}Bournemouth University \hspace{1cm} \textsuperscript{3}Technical University of Munich

Figure 1: From an incomplete point cloud ($N \times 3$) of a 3D scene (left), our method learns to jointly understand the 3D objects with semantic labels, poses (middle) and complete object meshes (right).

Abstract

Semantic scene understanding from point clouds is particularly challenging as the points reflect only a sparse set of the underlying 3D geometry. Previous works often convert point cloud into regular grids (e.g., voxels or bird-eye view images), and resort to grid-based convolutions for scene understanding. In this work, we introduce RfD-Net that jointly detects and reconstructs dense object surfaces directly from raw point clouds. Instead of representing scenes with regular grids, our method leverages the sparsity of point cloud data and focuses on predicting shapes that are recognized with high objectness. With this design, we decouple the instance reconstruction into global object localization and local shape prediction. It not only eases the difficulty of learning 2-D manifold surfaces from sparse 3D space, the point clouds in each object proposal convey shape details that support implicit function learning to reconstruct any high-resolution surfaces. Our experiments indicate that instance detection and reconstruction present complementary effects, where the shape prediction head shows consistent effects on improving object detection with modern 3D proposal network backbones. The qualitative and quantitative evaluations further demonstrate that our approach consistently outperforms the state-of-the-arts and improves over 11 of mesh IoU in object reconstruction.

1. Introduction

Semantic scene reconstruction has received increasing attention in applications such as robot navigation and interior design. It focuses on recovering the object labels, poses and geometries of objects in a 3D scene from partial observations (e.g., images or 3D scans). With the advance of 2D CNNs, instance reconstruction from images achieves appealing results \cite{20, 45, 36, 59, 33} but still bottlenecked by the depth ambiguity thus resulting in defective object locations. Compared to images, point clouds provide surface geometry that largely alleviates the object locating issues \cite{49, 2, 65, 7, 31}. However, its inherent sparseness and irregularity challenge the direct usage of grid-based CNNs on point clouds for semantic instance reconstruction.

Scanning 3D scenes usually results in missing geometries due to occlusions, view constraints and weak illumination where individual objects cannot be covered in all views. Prior works have explored various strategies to recover the missing shapes, e.g., depth inpainting, voxel/TSDF prediction, and shape retrieval. Depth inpainting \cite{23, 47, 28, 55} aims to complete the depth maps in single views. With 2D CNNs, these methods achieve pleasing results in surface points recovery. To complete a scene with occluded contents, many methods extend the 2D CNNs to 3D and naturally represent 3D scenes with voxel/TSDF grids \cite{57, 63, 37, 16, 13, 26, 43, 14}. This strategy also enables decoding voxel labels to complete scenes at the semantic or instance level. However, the expensive 3D convolutions in the scene level make them suffer from the resolution problem. Shape retrieval \cite{3, 4, 5, 30} provides an alternative method to predict shapes by searching for a CAD model as similar to the incomplete object as possible. However, the accuracy and the computation efficiency depends on the model dataset scale. Compared to voxels/TSDFs, point clouds present sparsity and are more scalable for efficient learning. To

\* Corresponding Email: hanxiaoguang@cuhk.edu.cn
our knowledge, few works have attempted to learn object meshes at the semantic-instance level directly from points.

In this work, we provide a reconstruction-from-detection framework, **RfD-Net**, for end-to-end semantic instance reconstruction directly from raw point clouds (see Figure 1). Our design is based on the insight that object detections provide spatial alignment that enables better local shape reconstruction. On the other hand, object shapes in sparse point clouds indicate local geometry that should back improve 3D detection. It decouples the problem of scene reconstruction into global localization and local shape prediction. Our method embeds the shape reconstruction head with a 3D detector backbone. It leverages the sparsity of point clouds and focuses on predicting shapes that are detected with high objectness and ignores the free scene space. With this design, our method allows implicit function learning to reconstruct surfaces with much higher resolution. In our experiments, we observe that joint shape reconstruction and 3D detection presents complementary effects. Deploying the shape prediction head is shown consistently effective on improving the modern point-based detectors, and vice versa, which makes our method consistently outperforms the prior arts in 3D detection and instance reconstruction. In summary, we list our contributions as follows:

- We provide a novel learning modality for semantic instance reconstruction. To our knowledge, it is the first learning method to predict instance semantics with geometry directly from point clouds. While prior methods heavily rely on 3D CNNs to learn from voxelized scenes.

- We propose a new end-to-end architecture, namely **RfD-Net**, to learn object semantics and shapes from sparse point clouds. It disentangles semantic instance reconstruction into global object localization and local shape prediction, which are bridged with a skip propagation module to facilitate joint learning. With this manner, our shape generator supports implicit learning, which directly overcomes the resolution bottleneck in the prior art [26].

- Joint learning object poses and shapes presents complementary benefits. It also shows consistent effects on modern detection backbones, and makes our method achieve the state-of-the-art in instance detection & completion and improve over 11 of mesh IoU in object reconstruction.

2. Related Work

In this section, we mainly summarize the recent works of 3D deep learning in shape completion, scene completion and instance-level scene reconstruction from point clouds.

**Shape Completion.** Shape completion aims to recover the missing geometries of an object from a partial scan. From point cloud inputs, many methods complete shapes represented by points [66, 58, 39, 46, 29, 61], voxels [15, 22], SDFs, [38, 48, 10, 42, 11] and meshes [21, 24]. Most of them share the similar modality, i.e., to encode the incomplete scan with point-wise convolutions (e.g., PointNet or PointNet++) and predict the invisible parts, holes or defective surfaces, while preserving the shape topology. They more focus on the completeness of single targets. In this work, we take advantage of the recent advance in shape completion for scene reconstruction. It supports our method to reconstruct high-resolution objects in a 3D scene.

**Scene Completion.** Scene completion focuses on predicting all objects including visible and invisible geometries from an incomplete scan. Different from shape completion, occlusions between objects manifest the major challenge. To this end, some works attempt to inpaint depth frames in scanning to recover object surfaces [23, 47, 28, 55]. Benefited from 2D CNNs, these methods achieve impressive results in surface points recovery. Similarly, more works extend the advantage of CNNs to 3D [19, 16, 13]. They discretize incomplete scans into voxel or TSDF grids and predict per-voxel occupancy with 3D CNNs for scene completion. With the fully convolution design, it also supports scene understanding tasks joint with completion, such as semantic completion or segmentation [57, 67, 35, 40, 63, 60, 34, 37, 9, 26, 14]. However, the expensive 3D CNNs consume much computation that hinders them with limited resolution, which appears more obvious on instance objects. In our work, we directly learn from point clouds. Objects detected with high objectness are only considered, which enables us to predict meshes with much higher resolution.

**Instance Reconstruction.** Beyond recovering scene geometries. Instance reconstruction refers to both object localization and reconstruction. With the advent of 3D scanners, early works [44, 56, 32, 8] focus on an approximate solution, i.e. semantic modeling. Object shapes are retrieved with CAD models or primitives with non-linear optimization. After that, 3D deep learning reforms this process into a learnable manner that model retrieval can be replaced with deep feature matching [3, 4, 5, 30]. Semantic modeling presents delicate shape models though, the matching similarity and inference efficiency directly rely on the CAD dataset scale. The closest topic to us is semantic instance reconstruction (or completion). [43] explores the possibility of predicting shapes along with object detection from voxelized point clouds. The previous state-of-the-art [26] discretizes the 3D scan into TSDF grids and predicts the occupancy of semantic instances with 3D CNNs. As we mentioned, their scene resolution (∼96×48×96) is limited by the heavy 3D CNN computation. Differently, we leverage the sparsity of point clouds where objects detected with high objectness will be reconstructed. It supports single object reconstruction to recover high-resolution shapes and jointly improves 3D object detection performance.
Figure 2: Overview of the network architecture. From an input point cloud with \( N \) points, a 3D proposal network backbone proposes \( N_p \) object proposals (\( D_p \)-dim features) that are decoded into \( D_b \)-dim box parameters. With a Top-\( N \) rank dropout layer, we reserve \( N_d \) proposals with higher objectness in the sparse point cloud. This subset of proposals are considered to independently group & align the neighboring \( M_p \) point cloud into clusters. Each point cluster is encoded into a \( D_s \)-dim vector through our point encoder to regress the binary occupancies of spatial query points (\( x-y-z \)) for mesh generation.

3. Method

We illustrate the architecture of RFD-Net in Figure 2. Our method follows the basic principle of understanding 3D scenes with ‘reconstruction from detection’. On this top, we devise the network consisting of a 3D detector, a spatial transformer and a shape generator. We build this architecture as generic as possible for learning instance shapes from point clouds, which should be flexibly compatible to modern point-based 3D proposal network backbones [49, 65]. Specifically, from an input point cloud, the 3D detector generates box proposals to locate object candidates from the sparse 3D scene. Then, we design a spatial transformer to select positive box proposals and group & align the local point cloud within for the next object shape generation. Shape generator independently learns an occupancy function in a canonical system to represent the shape of proposals. We elaborate on the details of each module as follows.

3.1. Learning Object Proposals in Point Clouds

From the input point cloud \( P \in \mathbb{R}^{N \times 3} \), we adopt the VoteNet [49] as the backbone to propose \( N_p \) box proposals with \( D_p \)-dim features \( F_p \in \mathbb{R}^{N_p \times D_p} \). We use \( F_p \) to predict \( D_b \)-dim box parameters, including center \( c \in \mathbb{R}^3 \), scale \( s \in \mathbb{R}^3 \), heading angle \( \theta \in \mathbb{R} \), semantic label \( l \) and objectness score \( s_{\text{obj}} \) (as parameterized in [49]). The objectness score is to classify if a proposal is close to (<0.3 m, positive) or far from (>0.6 m, negative) any ground-truth object center. We regress the box parameters with a 2-layer MLP.

In this part, each proposal feature \( f_p \) in \( F_p \) summarizes the semantics and box geometry information of a local region, which is of higher-level compared to the raw input points. In our method, \( F_p \) are also used to extend the local points for shape generation with our skip propagation module (see Section 3.3). It enhances the gradient back-propagation from shape generation to object detection, and is shown consistently effective on improving the object detection performance with different 3D detection backbones (see analysis experiments in section 5.3).

3.2. Transforming Points to Local

Objectness Dropout. To obtain the object points for shape generation, we develop a spatial transformer to group the input point cloud in box proposals, where object points in each proposal are transformed and aligned into a local canonical system (see Figure 2). Before the group & align process, we only reserve \( N_d \) positive boxes out of all the proposals during training. We adopt Top-\( N \) rank dropout to reserve \( N_d \) proposals with higher objectness (\( N_d = 10 \)). Different from images, 3D point cloud presents severe sparsity that free space occupies the major area. Too many negative samples (e.g., boxes with no/few points or far from any object) will be involved if considering all box proposals, which prevents the shape generator from learning correct shapes. Besides, the large computation also undermines the method efficiency. The effects with different numbers of positive proposals are discussed in our experiments. During inference, we replace the objectness dropout layer with a 3D NMS module [49] to produce the output boxes.

Group & Align. Within the \( N_d \) positive box proposals, we sample points from the input point cloud. We group points that are located within a radius \( r \) to these box centers \( \{c_i\} \) using a group layer [52]. It produces \( N_d \) clusters and \( M_p \) in-cluster points for each, where we denote the point clusters by \( \{P_i^c\} \) \((i = 1, 2, ..., N_d, P_i^c \in \mathbb{R}^{M_p \times 3}\)). Afterwards, we align the points in different clusters into a canonical coordinate system. It is to normalize the input points, which
removes the variance in spatial translation and rotation for shape prediction. We formulate this process by:

\[
P^c_i = \left[ R^{-1} (\theta_i) + \Delta R \right] \cdot \left[ P^c_i - (c_i + \Delta c) \right],
\]

where \( P^c_i \) denotes the aligned point cluster, \( c_i \) and \( \theta_i \) represents the corresponding box center and heading angle predicted from the 3D detector. \( R (\cdot) \in \mathbb{R}^{3 \times 3} \) is the rotation matrix. Since there would be deviations between the predicted centers & heading angles and the corresponding ground-truths, we learn an adjustment \((\Delta R, \Delta c)\) from the input point cluster \( P^c_i \) by:

\[
[\Delta R, \Delta c] = \text{MLP}_2 \left\{ \max_{p \in P^c_i} \{ \text{MLP}_1 (p) \} \right\}.
\]

In Equation 2, 3D points in each cluster \( P^c_i \) are independently processed with \( \text{MLP}_1 \), and max-pooled point-wisely into a global feature vector before being regressed into \((\Delta R, \Delta c)\) by \( \text{MLP}_2 \). So far, we have obtained the aligned 3D points \( \{ P^c_i \} \) with the corresponding proposal features \( \{ f_p \} \subset F_p \) of the positive box proposals.

### 3.3. Shape Generation from Proposals

We illustrate the shape generation module in Figure 3. Our shape generator consists of two parts: a skip propagation encoder to extend proposal features \( \{ f_p \} \) with cluster points \( \{ P^c_i \} \); a shape decoder to learn the occupancy of spatial points conditioned on the extended proposal features.

**Skip Propagation.** From the 3D points in proposals \( \{ P^c_i \} \), we first deploy a denoiser to reserve the foreground points for shape generation. It is implemented with PointNet [51] layers to classify if a point belongs to the foreground. We remove these background points with a point-wise ReLU layer. Foreground points in each proposal are afterward concatenated with the corresponding proposal feature \( f_p \) from the 3D detector by a skip connection. We propagate the upstream proposal feature to each 3D point, since it has summarized the information of object category and 3D size. We use it as shape priors to inform the decoder for efficient shape approximation. The skip propagation module bridges the gradient back-propagation from the shape generator to 3D detector and jointly improves shape prediction with object detection (see ablations in section 5.2). From the extended point clusters, we further deploy a PointNet [51] designed with residual connections to encode them into a set of new proposal features \( \{ f^*_p \in \mathbb{R}^{D_f} \} \).

**Shape Decoder.** We implicitly represent a 3D shape as an occupancy function [42, 11, 54]. That is, from spatial points \( \{ p \in \mathbb{R}^3 \} \) sampled in the shape bounding cube, we learn to predict their binary occupancy value \( o \in \{ 0, 1 \} \) (inside/outside the shape) conditioned on the input observation \( \{ f^*_p \} \). As in [42], we adopt the Conditional Batch Normalization [17, 18] layers to make the decoder conditioned on the proposal features to regress occupancy values. Furthermore, since the object point clouds are usually partially scanned, they could have multiple shape explanations. To this end, we build this decoder as a probabilistic generative model [64, 1] (see Figure 4). Specifically, from the input points \( \{ p \} \), occupancies \( \{ o \} \) and the proposal feature \( f^*_p \), we adopt the latent encoder in [42] and predict the mean and standard deviation \((\mu, \sigma)\) to approximate the standard normal distribution. We sample on our distribution to produce a latent code \( z \in \mathbb{R}^L \). \( \{ p \} \) and \( z \) are then processed into equal dimension with a single layer of MLP before summed point-wisely, and fed to five conditional blocks to regress the occupancies \( \{ o \} \). During inference, the latent code \( z \) is set to zeros, and we use the Marching Cubes algorithm [41] to extract the mesh surface from spatial occupancy grids.

### 4. End-to-end Learning

In this section, we summarize the learning targets with corresponding losses for end-to-end training.
**Box Loss.** The 3D detector predicts the proposal objectness score \( s_{obj} \), box center \( c \), scale \( s \), heading angle \( \theta \) with the semantic label \( l \). As in [49], we supervise the objectness loss \( \mathcal{L}_{obj} \) of box proposals with cross entropy, to classify if they are located within 0.3 meters (positive) or more than 0.6 meters (negative) to any object center. Proposals with positive objectness are supervised with the following box loss. We supervise the box center loss \( \mathcal{L}_c \) using Smooth-L1 loss, and follow [50, 27] to disentangle the scale loss \( \mathcal{L}_s \) and the heading angle loss \( \mathcal{L}_\theta \) into a hybrid of classification (cross entropy) and regression (Smooth-L1) losses, i.e., \( \lambda_{cls} \mathcal{L}_{cls} + \mathcal{L}_{reg} \). We use the cross-entropy as the semantic classification loss \( \mathcal{L}_l \). Each proposal is paired with the nearest ground-truth box for training. For using VoteNet as the backbone, we involve an extra vote loss \( \mathcal{L}_v \) as in [49]. So far the above losses can be concluded into the box loss \( \mathcal{L}_{box} \) with weighted sum.

**Shape Loss.** For points in each proposal, we supervise the foreground segmentation loss \( \mathcal{L}_{seg} \) with cross entropy. The shape generator learns \((\mu, \sigma)\) to approximate the standard normal distribution in training, from which we sample a latent code \( z \) to predict the occupancy values \( \{ o \} \) of spatial query points \( \{ p \} \) with a condition of the proposal feature \( f_p \). Then the \( \mathcal{L}_{shape} \) can be formulated by

\[
\mathcal{L}_{shape} = \frac{1}{N_d} \sum_{i=1}^{N_d} \left[ \sum_{j=1}^{K} \mathcal{L}_{cc}(\hat{o}_{i,j}, o_{i,j}) \right] + \text{KL}(\hat{p}(z_i) \| p(z_i)) + \lambda_{seg} \mathcal{L}_{seg},
\]

where \( \mathcal{L}_{cc} \) and KL denotes the cross entropy loss and KL-divergence respectively. \( \hat{o}_{i,j} \) and \( o_{i,j} \) correspond to the predicted and ground-truth occupancy of the \( j \)-th spatial point in the \( i \)-th positive proposal. \( \hat{p}(z_i) \) is the predicted distribution of the latent code \( z_i \), \( p(z_i) \) is the target distribution, which is set to the standard normal distribution as [42].

Overall, we train our network end-to-end with the loss of \( \mathcal{L} = \mathcal{L}_{box} + \lambda \mathcal{L}_{shape} \). The weights to balance different losses are detailed in the supplementary material.

### 5. Results and Evaluation

#### 5.1. Experiment Setup

**Data.** Two datasets are used in our experiments. 1) ScanNet v2 [12] consists of 1,513 real world scans with point clouds labeled at the instance level. 2) Scan2CAD [3] aligns the ShapeNet [6] models with the object instances in ScanNet. It provides the object meshes. We preprocess these object meshes following [42] to prepare the spatial points and occupancy values for shape learning. We only use the point clouds from ScanNet as the input (randomly sampled with 80K points in train and test), and predict object bounding boxes and meshes supervised by Scan2CAD. Inline with [26], we use the official train/test split in all the experiments and consider eight object categories.

**Metrics.** We evaluate our method on both scene understanding and object reconstruction, including 3D detection, single object reconstruction, and semantic instance completion. As [26], the mean average precision at the 3D IoU threshold of 0.5 (mAP@0.5) is used in the evaluation of 3D detection and semantic instance completion, and we adopt 3D IoU for mesh evaluation in single object reconstruction.

**Implementation.** Our network is end-to-end trained before pretraining the 3D proposal network. We set the batch size as 8, and adopt the initial learning rate at 1e-3 in pretraining and 1e-4 in end-to-end training, which drops by the scale of 0.5 for every 80 epochs. 240 epochs are used in total. During inference, we replace the objectness dropout (see Section 3.1) with 3D NMS [49] to output the 3D boxes, where object meshes in these boxes are extracted from 128\(^3\) occupancy grids and reconstructed with Marching Cubes [41]. All object meshes are predicted in a canonical system and transformed to 3D scenes with the predicted box poses. We provide the full list of parameters, layer specifications, and efficiency&memory comparisons in the supplementary file.

**Benchmark.** We compare our method on the metrics above with the previous state-of-the-art works in scene reconstruction, including RevealNet [26], 3D-SIS [25] and ScanComplete [14]. We also investigate the compatibility of our method by embedding with different 3D proposal network backbones, including BoxNet [49], MLCVNet [65], and VoteNet with Graph Voting [62]. All results in our experiments are trained and tested with the same dataset and split. We keep the geometric data of 3D scans as the input in comparisons. For some methods requiring color images, we keep inline with their original configuration.

### 5.2. Comparisons to the State-of-the-Arts

**Qualitative Comparisons.** We compare our method on semantic instance completion with the first and currently the state-of-the-art work, RevealNet [26]. Figure 5 illustrates some qualitative results with different scene complexities on the test set of ScanNet (see more comparisons in the supplementary file). RevealNet discretizes the input scans into a volumetric grid that is preprocessed as a truncated signed distance field (TSDF). Their network is backboned with 3D CNN blocks to encode and decode the voxel occupancy of each object with a fully convolutional fashion. It can also back-project the 2D image features to TSDF grids to extend the representation of the input scans. In our method, only the geometry input is needed. The results in Figure 5 show that our method presents better quality of object boxes and shape details. The reasons could be three-fold: 1. different with RevealNet which processes the whole scene with 3D CNNs, our method leverages the sparsity of point cloud and focuses on the object proposals with higher objectness. It
enables us to ignore those empty spaces and saves computation load to reconstruct the objects of interest with high resolution (128-d); 2. voxelizing scenes into grids shows inadequacy in representing accurate object boxes (i.e., discrete box coordinates with axis-aligned orientations); 3. our shape generator presents joint effect that further improves the 3D detector and achieves better object localization.

We further quantitatively evaluate our method on single tasks as follows. Two configurations are considered to explore the complementary effects between the 3D detection and shape generation modules, i.e., training our network end-to-end (w/ joint) or training the 3D detector and shape generator individually with the other one fixed (w/o joint).

**3D Object Detection.** We evaluate our 3D detection by comparing with the prior art works, 3D-SIS [25], MLCVNet [65] and RevealNet [26] (see Table 2). 3D-SIS fuses the multi-view RGB images into the TSDF grids (by back-projecting 2D CNN features to 3D voxels) as inputs. It adapts the detection modality of Faster RCNN [53] from 2D image plane to 3D voxel grids. MLCVNet [65] extends VoteNet [49] by considering contextual information between objects. From the results, we observe that, with joint training, our shape generation module improves the 3D proposal network backbone (i.e., w/o joint training) on all categories (see per category scores in the supplementary file), and helps our method outperforms the state-of-the-arts, which is also shown consistently effective on other backbones (see analysis experiments).
Table 1: Comparisons on semantic instance completion. The results of (Inst Seg [25] + Shape Comp [15]) and (Scan Comp [14] + Inst Seg [25]) are provided by [26]. The mAP scores are measured with the mesh IoU threshold at 0.5.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RevealNet [26]</td>
<td>Geo Only</td>
<td>29.29</td>
</tr>
<tr>
<td>Ours (w/o joint)</td>
<td>Geo Only</td>
<td>32.63</td>
</tr>
<tr>
<td>Ours (w/ joint)</td>
<td>Geo Only</td>
<td><strong>35.10</strong></td>
</tr>
</tbody>
</table>

Table 2: 3D object detection on ScanNet v2. 3D-SIS [25] and RevealNet [26] results are provided by the authors. MLCVNet results are retrained with the original network [65]. See per category scores in the supplementary file.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-SIS [25]</td>
<td>Geo+Image</td>
<td>25.70</td>
</tr>
<tr>
<td>MLCVNet [65]</td>
<td>Geo Only</td>
<td>33.40</td>
</tr>
<tr>
<td>RevealNet [26]</td>
<td>Geo Only</td>
<td>29.29</td>
</tr>
<tr>
<td>Ours (w/o joint)</td>
<td>Geo Only</td>
<td>32.63</td>
</tr>
<tr>
<td>Ours (w/ joint)</td>
<td>Geo Only</td>
<td><strong>35.10</strong></td>
</tr>
</tbody>
</table>

Table 3: Comparisons on object reconstruction. Per category scores are detailed in the supplementary file.

<table>
<thead>
<tr>
<th>Method</th>
<th>resolution</th>
<th>3D IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>RevealNet [26]</td>
<td>avg. 27-d</td>
<td>20.48</td>
</tr>
<tr>
<td>Ours (w/ &amp; w/o joint)</td>
<td>16-d</td>
<td><strong>37.02</strong> &amp; <strong>35.75</strong></td>
</tr>
<tr>
<td>Ours (w/ &amp; w/o joint)</td>
<td>32-d</td>
<td><strong>31.81</strong> &amp; <strong>30.21</strong></td>
</tr>
<tr>
<td>Ours (w/ &amp; w/o joint)</td>
<td>64-d</td>
<td><strong>26.65</strong> &amp; <strong>24.97</strong></td>
</tr>
</tbody>
</table>

**Semantic Instance Completion.** We evaluate our method on scene completion at semantic-instance level. It measures how much the predicted object meshes overlay the ground-truths in a 3D scene. We compare our method with the state-of-the-art work, RevealNet [26]. We also follow their experiments by sequentially combing prior arts of instance segmentation [25] with shape completion [15], or combining scan completion ([14]) with instance segmentation [25], for instance-level scene completion. As mentioned above, object meshes in our method are predicted with a higher resolution (128-d). For a fair comparison, we voxelize our scenes to the same resolution with them. The results are listed in Table 1. It shows that our joint method largely outperforms the decoupled methods and exceeds the state-of-the-art [26]. The comparison between w/ & w/o joint training further explains the complementary effects between shape generation and 3D detection, which is inline with the former results that improving on each module will benefit the other one. For objects with very thin structure (e.g. tables), our method does not show advantage because a slight misalignment in 3D detection is more likely to affect their mesh overlay to the ground-truth in 3D scenes, although the object meshes are well reconstructed.

**5.3. Analysis Experiments**

**Different Backbones.** In our network, we deploy the backbone of our 3D proposal network using VoteNet [49]. We change our backbone to explore if the joint training still benefits other 3D detection networks. We consider in our experiments with the BoxNet [49], MLCVNet [65] and VoteNet+Dynamic Graph CNN (DGCCN) [62]. BoxNet directly predicts 3D boxes from the grouped points in point clouds. MLCVNet is a variant version of VoteNet that considers contextual information between objects. To this end, we also configure dynamic graph connections [62] between votes to learn the relational features for object detection. The results in Table 4 manifest that combing shape prediction in joint training consistently improves the 3D detection performance with different backbones.

**Different Point Sparsity.** Learning from point clouds makes our network adaptive to different scales of input points. We test the robustness of our method on different point sparsity, where our network is trained with 20K, 40K
and 80K input points. The results in Figure 6 and Table 5 show that our method still can produce reasonable scenes with very sparse inputs (20K).

<table>
<thead>
<tr>
<th>Backbone</th>
<th>20K</th>
<th>40K</th>
<th>80K</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxNet [49]</td>
<td>20.31</td>
<td>22.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLCVNet [65]</td>
<td>33.40</td>
<td>34.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VoteNet+DGCNN [62]</td>
<td>29.88</td>
<td>32.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>32.63</td>
<td>35.10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: 3D detection with different backbones.

6. Conclusion

We propose a novel learning modality, namely RfD-Net, for semantic instance reconstruction directly from point clouds. It disentangles the problem with a reconstruction-from-detection manner. Instance shapes are predicted with global object localization and local shape prediction, which are connected with a spatial transformer and a skip propagation module to bridge the information flow from shapes to detections. It facilitates the complementary effects and jointly improves the performance of 3D detection and shape generation. The experiments further demonstrate that our method achieves much better mesh quality in scene reconstruction and outperforms the state-of-the-art in object reconstruction, 3D detection and semantic instance completion.

Acknowledgements. This project was supported in part by the National Key R&D Program of China with grant No. 2018YFB1800800, the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, Guangdong Research Project No. 2017ZT07X152, the National Natural Science Foundation of China 61902334, Shenzhen Fundamental Research (General Project) JCYJ20190814112007258, a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant Scan2CAD (804724), the German Research Foundation (DFG) Grant Making Machine Learning on Static and Dynamic 3D Data Practical, Bournemouth University and China Scholarship Council. We also would like to thank Prof. Angela Dai for the video voice over in our work.

Different Objectness Thresholds. In section 3.2, we adopt Objectness Dropout to keep proposal boxes with higher objectness (Top-$N_d$). It ensures meaningful object points as the input to learn shapes from those positive proposals. Thus, a high $N_d$ will involve in negative proposals (e.g. far away from any object) and undermines the shape learning, while a low $N_d$ will reduce the learning efficiency. We list the effects of using different $N_d$ in Table 6.

Different Skip Propagation. In section 3.3, we propagate the proposal features $\{f_p\}$ to point clusters $\{P_i\}$ for shape generation. To investigate its effects, we configure extra three approaches of feature propagation for shape generation: 1. to decode shapes from proposal features without point clusters ($c_1$); 2. to learn shapes from the point clusters without proposal features ($c_2$); 3. to propagate the proposal features to point clusters but without the point denoizer ($c_3$). All these configurations are jointly trained. We list the comparisons in Table 7. The results indicate that the feature extraction for shape generation mainly affects the downstream shape generator thus further influence the instance completion, where combing the proposal features and point clusters with our skip propagation outcomes the best configuration.

<table>
<thead>
<tr>
<th>$N_d$</th>
<th>3D Detection</th>
<th>Instance Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>34.33</td>
<td>14.48</td>
</tr>
<tr>
<td>10</td>
<td>35.10</td>
<td>16.90</td>
</tr>
<tr>
<td>30</td>
<td>33.37</td>
<td>14.34</td>
</tr>
</tbody>
</table>

Table 6: Results on different thresholds for objectness dropout. $N_d=10$ is used in our network.

<table>
<thead>
<tr>
<th>$N_d$</th>
<th>3D Detection</th>
<th>Instance Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>c_1</td>
<td>34.83</td>
<td>13.97</td>
</tr>
<tr>
<td>c_2</td>
<td>34.85</td>
<td>14.33</td>
</tr>
<tr>
<td>c_3</td>
<td>35.03</td>
<td>15.42</td>
</tr>
<tr>
<td>Full</td>
<td>35.10</td>
<td>16.90</td>
</tr>
</tbody>
</table>

Table 7: Ablation on skip propagation.
References


[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification...
and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 652–660, 2017. 2, 4


[64] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In Advances in neural information processing systems, pages 82–90, 2016. 4

