

RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction

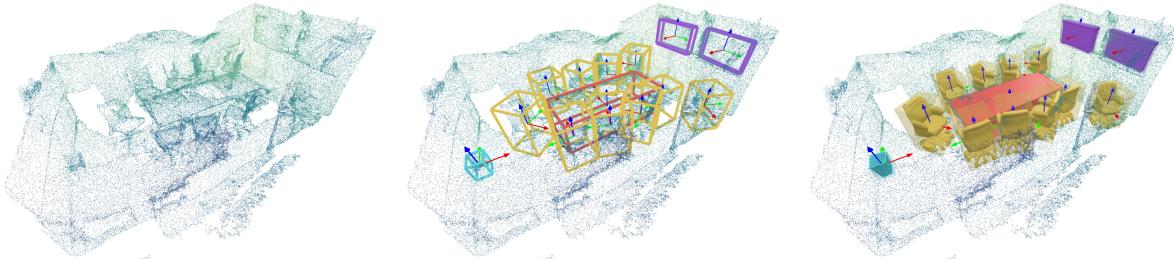
Yinyu Nie^{1,2}¹SRIBD, CUHK SZJi Hou³²Bournemouth UniversityXiaoguang Han^{1,*}Matthias Nießner³

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 [19, 44, 35, 57, 32] 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 [47, 2, 63, 7, 30]. 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 [22, 46, 27, 53] 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 [55, 61, 36, 15, 12, 25, 42, 13]. 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 [3, 4, 5, 29] provides an alternative method to pre-

* Corresponding Email: hanxiaoguang@cuhk.edu.cn

dict 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 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 [25].
- 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 a target object from a partial scan. From point cloud inputs, many methods complete shapes represented by points [64, 56, 38, 45, 28, 59], voxels [14, 21], SDFs, [37, 41, 10] and meshes [20, 23]. Most of them share the similar modality, i.e., to encode the incomplete scan with point-wise convolutions (e.g., PointNet or PointNet++ [49, 50]) 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 [22, 46, 27, 53]. Benefited from 2D CNNs, these methods achieve impressive results in surface points recovery. Similarly, more works extend the advantage of CNNs to 3D [18, 15, 12]. 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 [55, 65, 34, 39, 61, 58, 33, 36, 9, 25, 13]. 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 [43, 54, 31, 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, 29]. 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). [42] explores the possibility of predicting shapes along with object detection from voxelized point clouds. The previous state-of-the-art [25] discretizes the 3D scan into TSDF grids and predicts the occupancy of semantic instances with 3D CNNs. As we mentioned, their scene resolution ($\approx 96 \times 48 \times 96$) is limited by the heavy 3D CNN computation. Differently, we leverage

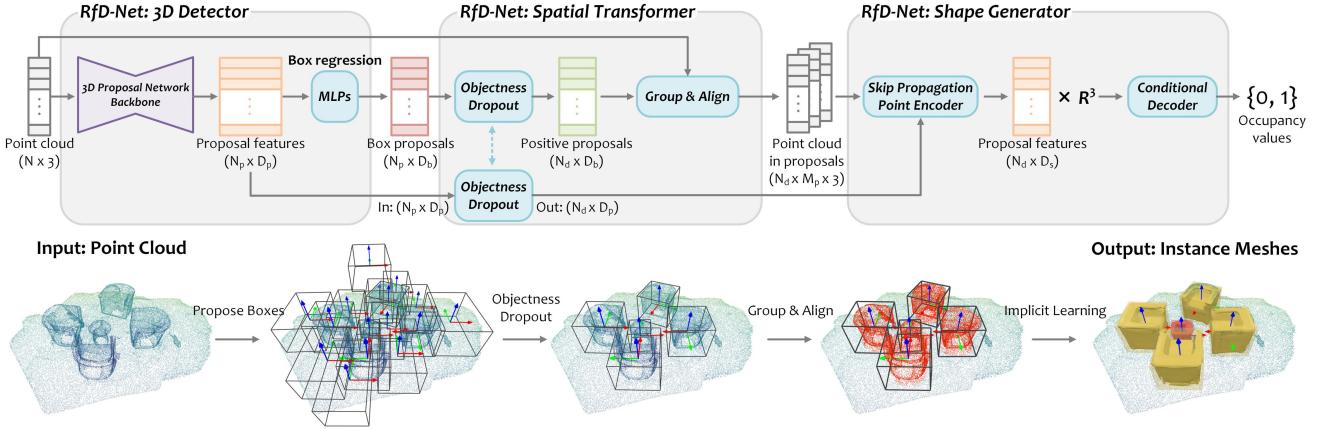


Figure 2: Overview of the network architecture. From an input point cloud with N points, a 3D proposal network backbone proposes N_p object proposals (D_p -dim features) that are decoded into D_b -dim box parameters. With a Top- N rank dropout layer, we reserve N_d proposals with higher objectness in 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.

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.

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 [47, 63]. 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 $\mathbf{P} \in \mathbb{R}^{N \times 3}$, we adopt the VoteNet [47] as the backbone to propose N_p box proposals with D_p -dim features $\mathbf{F}_p \in \mathbb{R}^{N_p \times D_p}$. We use \mathbf{F}_p to predict D_b -dim box parameters, including center $\mathbf{c} \in \mathbb{R}^3$, scale $\mathbf{s} \in \mathbb{R}^3$, heading angle $\theta \in \mathbb{R}$, semantic label l and objectness score s_{obj} (as parameterized in [47]). 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 \mathbf{f}_p in \mathbf{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, \mathbf{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 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 $\{\mathbf{c}_i\}$

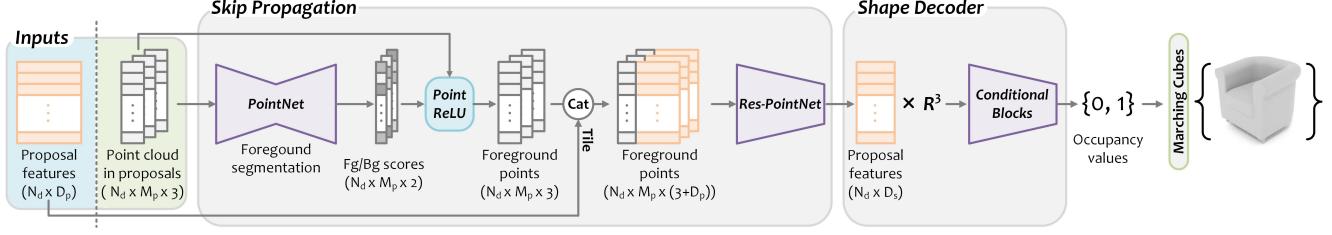


Figure 3: Shape generation from object proposals. From the 3D points in the positive N_d object proposals, our method first learns a denoiser (with PointNet [49] layers) to remove the background points. These cleaned points are afterwards extended by the former proposal box features with a skip connection, before being encoded into new proposal features to decode spatial occupancy values. The object meshes are obtained by Marching Cubes [40] on the predicted occupancy grids.

using a group layer [50]. It produces N_d clusters and M_p in-cluster points for each, where we denote the point clusters by $\{\mathbf{P}_i^c\}$ ($i = 1, 2, \dots, N_d, \mathbf{P}_i^c \in \mathbb{R}^{M_p \times 3}$). 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:

$$\tilde{\mathbf{P}}_i^c = [\mathcal{R}^{-1}(\theta_i) + \Delta\mathcal{R}] \cdot [\mathbf{P}_i^c - (\mathbf{c}_i + \Delta\mathbf{c})], \quad (1)$$

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

$$[\Delta\mathcal{R}, \Delta\mathbf{c}] = \text{MLP}_2 \left\{ \max_{p \in \mathbf{P}_i^c} \{\text{MLP}_1(p)\} \right\}. \quad (2)$$

In Equation 2, 3D points in each cluster \mathbf{P}_i^c are independently processed with MLP_1 , and max-pooled point-wisely into a global feature vector before being regressed into $(\Delta\mathcal{R}, \Delta\mathbf{c})$ by MLP_2 . So far, we have obtained the aligned 3D points $\{\tilde{\mathbf{P}}_i^c\}$ 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 $\{\tilde{\mathbf{P}}_i^c\}$; a shape decoder to learn the occupancy of spatial points conditioned on the extended proposal features.

Skip Propagation. From the 3D points in proposals $\{\tilde{\mathbf{P}}_i^c\}$, we first deploy a denoiser to reserve the foreground points for shape generation. It is implemented with PointNet [49] 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

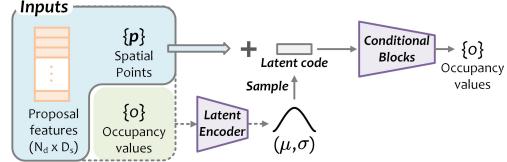


Figure 4: Probabilistic shape generation in training.

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 [49] designed with residual connections to encode them into a set of new proposal features $\{f_p^* \in \mathbb{R}^{D_s}\}$.

Shape Decoder. We implicitly represent a 3D shape as an occupancy function [41, 10, 52]. 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/ouside the shape) conditioned on the input observation (f_p^*) . As in [41], we adopt the Conditional Batch Normalization [16, 17] 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 [62, 1] (see Figure 4). Specifically, from the input points $\{p\}$, occupancies $\{o\}$ and the proposal feature f_p^* , we adopt the latent encoder in [41] and predict the mean and standard deviation (μ, σ) 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 [40] 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 θ with the semantic label l . As in [47], 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- L_1 loss, and follow [48, 26] to disentangle the scale loss \mathcal{L}_s and the heading angle loss \mathcal{L}_θ into a hybrid of classification (cross entropy) and regression (Smooth- L_1) 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 [47]. 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 (μ, σ) 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

$$\begin{aligned} \mathcal{L}_{shape} = & \frac{1}{N_d} \sum_{i=1}^{N_d} \left[\sum_{j=1}^K \mathcal{L}_{ce}(\hat{o}_{i,j}, o_{i,j}) \right. \\ & \left. + \text{KL}(\hat{p}(z_i) \| p(z_i)) \right] + \lambda_{seg}\mathcal{L}_{seg}, \end{aligned} \quad (3)$$

where \mathcal{L}_{ce} 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 [41].

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 [11] 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 [41] 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 [25], 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 [25], 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 at 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 [47] to output the 3D boxes, where object meshes in these boxes are extracted from 128^3 occupancy grids and reconstructed with Marching Cubes [40]. 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 [25], 3D-SIS [24] and ScanComplete [13]. We also investigate the compatibility of our method by embedding with different 3D proposal network backbones, including BoxNet [47], MLCVNet [63], and VoteNet with Graph Voting [60]. 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 [25]. 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

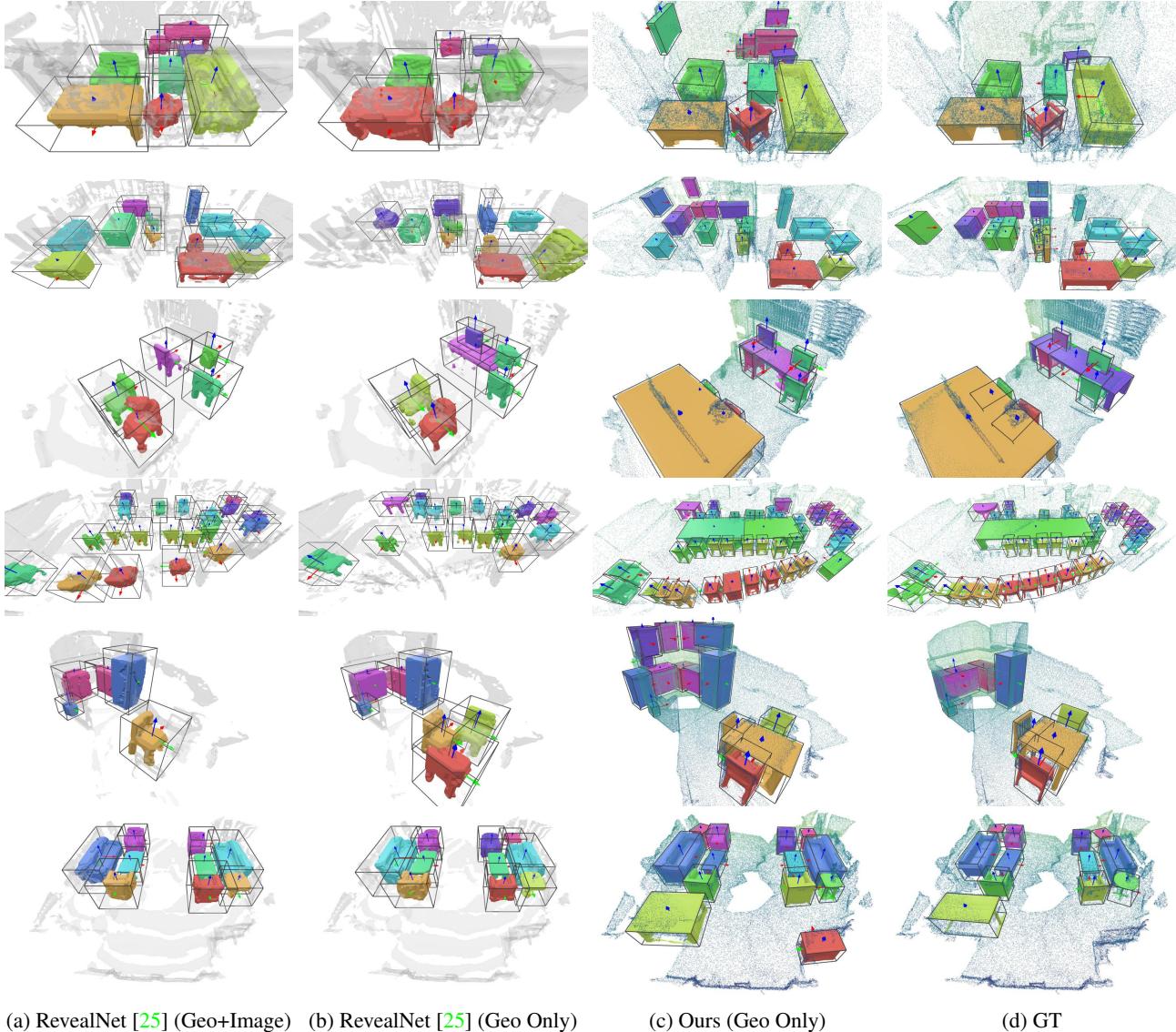


Figure 5: Qualitative results of semantic instance reconstruction on ScanNet v2 [11]. Note that RevealNet [25] preprocesses the scanned scenes into TSDF grids, while our method only uses the raw point clouds.

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 [24], MLCVNet [63] and RevealNet [25] (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 [51] from 2D image plane to 3D voxel grids. MLCVNet [63] extends VoteNet [47] 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 supplemen-

	display	bathtub	trashbin	sofa	chair	table	cabinet	bookshelf	mAP
Inst Seg [24] + Shape Comp [14]	2.27	1.14	1.68	14.86	9.93	3.90	7.11	3.03	5.49
Scan Comp [13] + Inst Seg [24]	1.65	4.55	11.25	9.09	9.09	0.64	0.18	5.45	5.24
RevealNet [25]	13.16	13.64	18.19	24.79	15.87	11.28	8.60	10.60	14.52
Ours (w/o joint)	23.96	19.00	18.49	13.67	10.84	2.81	10.14	12.65	13.94
Ours (w/ joint)	26.67	27.57	23.34	15.71	12.23	1.92	14.48	13.39	16.90

Table 1: Comparisons on semantic instance completion. The results of (Inst Seg [24] + Shape Comp [14]) and (Scan Comp [13] + Inst Seg [24]) are provided by [25]. The mAP scores are measured with the mesh IoU threshold at 0.5.

	Input	mAP
3D-SIS [24]	Geo+Image	25.70
MLCVNet [63]	Geo Only	33.40
RevealNet [25]	Geo Only	29.29
Ours (w/o joint)	Geo Only	32.63
Ours (w/ joint)	Geo Only	35.10

Table 2: 3D object detection on ScanNet v2. 3D-SIS [24] and RevealNet [25] results are provided by the authors. MLCVNet results are retrained with the original network [63]. See per category scores in the supplementary file.

tary file), and helps our method outperforms the state-of-the-arts, which is also shown consistently effective on other backbones (see analysis experiments).

Object Reconstruction. We further evaluate the single object reconstruction quality with 3D mesh IoU, where the object shapes are predicted and evaluated in a canonical system (disentangled with 3D detection). Note that all the predictions and groundtruths in RevealNet [25] share the same coordinate system, i.e. the TSDF grid. RevealNet voxelizes 3D scenes with a uniform voxel size of ≈ 4.7 cm, which results in around $96 \times 48 \times 96$ voxels for a scene at the scale of $4.5 \times 2.25 \times 4.5$ meters. However, objects are independently reconstructed with the same resolution in our method. For a fair comparison, we present our results on different resolutions (16-d, 32-d, 64-d) and benchmark all methods with only geometric input (see the comparisons in Table 3). The results indicate that joint training the 3D detector with shape generator can even improve the performance on single object reconstruction. It implies that better box proposals will produce better spatial alignment and meaningful descriptions about the object property (e.g., size and category), which informs the shape generator to better approximate the target shapes. The results also demonstrate our mesh quality outperforms RevealNet with a large margin (over 11 points). Note that the resolution of 64-d even exceeds their scene resolution in vertical axis (48-d [25]).

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

	resolution	3D IoU
RevealNet [25]	avg. 27-d	20.48
Ours (w/ & w/o joint)	16-d	37.02 & 35.75
Ours (w/ & w/o joint)	32-d	31.81 & 30.21
Ours (w/ & w/o joint)	64-d	26.65 & 24.97

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

state-of-the-art work, RevealNet [25]. We also follow their experiments by sequentially combining prior arts of instance segmentation [24] with shape completion [14], or combining scan completion ([13]) with instance segmentation [24], 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 (4.7 cm per voxel) 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 [25]. 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 [47]. We change our backbone to explore if the joint training still benefits other 3D detection networks. We consider in our experiments with the BoxNet [47], MLCVNet [63] and VoteNet+Dynamic Graph CNN (DGCNN) [60]. 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 [60] between votes to learn the relational features for object detection. The results in Table 4 manifest that combining shape predic-

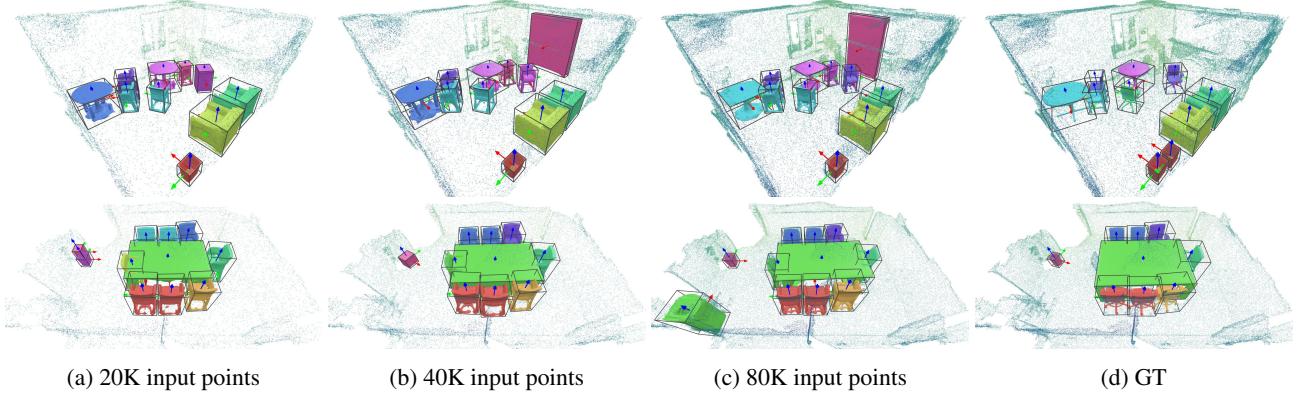


Figure 6: Scene reconstruction with different point sparsity.

tion in joint training consistently improves the 3D detection performance with different backbones.

	w/o joint	w/ joint
BoxNet [47]	20.31	22.79
MLCVNet [63]	33.40	34.77
VoteNet+DGCNN [60]	29.88	32.30
Ours	32.63	35.10

Table 4: 3D detection 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 scan produce reasonable scenes with very sparse inputs (20K).

	3D Detection	Instance Completion
20K	33.89	14.39
40K	34.75	15.23
80K	35.10	16.90

Table 5: Comparisons with different scales of input points.

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 $\{\hat{P}_i^c\}$ 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

N_d	3D Detection	Instance Completion
5	34.33	14.48
10	35.10	16.90
30	33.37	14.34

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

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 combining the proposal features and point clusters with our skip propagation outcomes the best configuration.

	3D Detection	Instance Completion
c_1	34.83	13.97
c_2	34.85	14.33
c_3	35.03	15.42
Full	35.10	16.90

Table 7: Ablation on skip propagation.

6. Conclusion

We propose a novel learning modality, namely *RfD-Net*, for semantic instance reconstruction directly 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.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. [4](#)
- [2] Syeda Mariam Ahmed and Chee Meng Chew. Density-based clustering for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10608–10617, 2020. [1](#)
- [3] Armen Avetisyan, Manuel Dahner, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019. [1, 2, 5, 13](#)
- [4] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2551–2560, 2019. [1, 2](#)
- [5] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. *arXiv preprint arXiv:2003.12622*, 2020. [1, 2](#)
- [6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [5](#)
- [7] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 392–401, 2020. [1](#)
- [8] Kang Chen, Yu-Kun Lai, Yu-Xin Wu, Ralph Martin, and Shi-Min Hu. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics*, 33(6), 2014. [2](#)
- [9] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4193–4202, 2020. [2](#)
- [10] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. [2, 4](#)
- [11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. [5, 6, 13, 14](#)
- [12] Angela Dai, Christian Diller, and Matthias Nießner. Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2020. [1, 2](#)
- [13] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. [1, 2, 5, 7](#)
- [14] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. [2, 7](#)
- [15] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. *arXiv preprint arXiv:2006.14660*, 2020. [1, 2](#)
- [16] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017. [4, 12](#)
- [17] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. [4, 12](#)
- [18] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. [2](#)
- [19] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019. [1](#)
- [20] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. [2](#)
- [21] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1029, 2019. [2](#)

- ternational Conference on Computer Vision*, pages 85–93, 2017. 2
- [22] Xiaoguang Han, Zhaoxuan Zhang, Dong Du, Mingdai Yang, Jingming Yu, Pan Pan, Xin Yang, Ligang Liu, Zixiang Xiong, and Shuguang Cui. Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2019. 1, 2
- [23] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. *arXiv preprint arXiv:2005.11084*, 2020. 2
- [24] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 5, 6, 7, 15
- [25] Ji Hou, Angela Dai, and Matthias Nießner. Revealnet: Seeing behind objects in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2020. 1, 2, 5, 6, 7, 13, 14, 15
- [26] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In *Advances in Neural Information Processing Systems*, pages 207–218, 2018. 5
- [27] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H Hsu. Indoor depth completion with boundary consistency and self-attention. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2
- [28] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 2
- [29] Vladislav Ishmtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Niessner, Denis Zorin, and Evgeny Burnaev. Cad-deform: Deformable fitting of cad models to 3d scans. *arXiv preprint arXiv:2007.11965*, 2020. 1, 2
- [30] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020. 1
- [31] Young Min Kim, Niloy J Mitra, Dong-Ming Yan, and Leonidas Guibas. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. 2
- [32] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568, 2018. 1
- [33] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3351–3359, 2020. 2
- [34] Jie Li, Yu Liu, Xia Yuan, Chunxia Zhao, Roland Siegwart, Ian Reid, and Cesar Cadena. Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters*, 5(1):219–226, 2019. 2
- [35] Kejie Li, Martin Rünz, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. *arXiv preprint arXiv:2005.05125*, 2020. 1
- [36] Siqi Li, Changqing Zou, Yipeng Li, Xibin Zhao, and Yue Gao. Attention-based multi-modal fusion network for semantic scene completion. *arXiv preprint arXiv:2003.13910*, 2020. 1, 2
- [37] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 2
- [38] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11596–11603, 2020. 2
- [39] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li. See and think: Disentangling semantic scene completion. In *Advances in Neural Information Processing Systems*, pages 263–274, 2018. 2
- [40] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 4, 5, 12
- [41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2, 4, 5, 12
- [42] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. Dops: Learning to detect 3d objects and predict their 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11913–11922, 2020. 1, 2
- [43] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012. 2
- [44] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 1
- [45] Yinyu Nie, Yiqun Lin, Xiaoguang Han, Shihui Guo, Jian Chang, Shuguang Cui, and Jian Jun Zhang. Skeleton-bridged point completion: From global inference to local adjustment. *arXiv preprint arXiv:2010.07428*, 2020. 2
- [46] David Novotny, Ben Graham, and Jeremy Reizenstein. Perspectivenet: A scene-consistent image generator for new view synthesis in real indoor environments. In *Advances in Neural Information Processing Systems*, pages 7601–7612, 2019. 1, 2

- [47] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 3, 5, 6, 7, 8, 12
- [48] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgbd data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 5
- [49] 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, 12
- [50] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 4, 12
- [51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6
- [52] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morigima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. 4
- [53] Dmitry Senushkin, Ilia Belikov, and Anton Konushin. Decoder modulation for indoor depth completion. *arXiv preprint arXiv:2005.08607*, 2020. 1, 2
- [54] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baineng Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. 2
- [55] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017. 1, 2
- [56] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 2
- [57] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 302–310, 2018. 1
- [58] Peng-Shuai Wang, Yang Liu, and Xin Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 266–267, 2020. 2
- [59] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 2
- [60] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 5, 7, 8
- [61] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8608–8617, 2019. 1, 2
- [62] 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
- [63] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10447–10456, 2020. 1, 3, 5, 6, 7, 8, 15
- [64] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 2
- [65] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 733–749, 2018. 2

Appendix

A. Network and Layer Specifications

In this section, we provide all the parameters, layer specifications and weights used in loss functions. We uniformly denote the fully-connected layers by $\text{MLP}[l_1, \dots, l_d]$, where l_i is the number of neurons in the i -th layer.

A.1. 3D Detector

In section 3.1, we predict object proposals from N input points with VoteNet [47] as the backbone. It produces N_p proposals with D_p -dim features (i.e. proposal features $\mathbf{F}_p \in \mathbb{R}^{N_p \times D_p}$ in our paper), from which we regress the D_b -dim box parameters with $\text{MLP}[128, 128, 69]$ ($N=80\text{K}$, $N_p=256$, $D_p=128$, $D_b=69$). As in [47], the 69-dim box parameters are encoded by center $\mathbf{c} \in \mathbb{R}^3$, scale $\mathbf{s}^3 \in \mathbb{R}^3$, heading angle $\theta \in \mathbb{R}$, semantic label l , and objectness score s_{obj} . s_{obj} is a probability value indicating whether the proposal is close to (<0.3 meter, positive) or far from (>0.6 meter, negative) any ground-truth object center.

A.2. Spatial Transformer

Objectness Dropout Layer. In section 3.2, we adopt an *Objectness Dropout* layer to reserve the Top- N_d proposals with higher objectness ($N_d = 10$) for shape learning. In test, we replace it with 3D Non Maximum Suppression (3D NMS) to produce the output boxes and corresponding shapes with the 3D Box IoU threshold of 0.25 in evaluation.

Group & Align. From the N_d box proposals, we group the neighboring M_p points that are located within a radius r to each box center using a group layer [50]. $M_p=1024$, $r=1$. It produces N_d point clusters $\{\mathbf{P}_i^c\}$ ($i = 1, 2, \dots, N_d$, $\mathbf{P}_i^c \in \mathbb{R}^{M_p \times 3}$). After grouping, we align the 3D points in each cluster to a canonical system with the Equation 1 in our paper, where the rotation and translation adjustment ($\Delta\mathcal{R}, \Delta\mathcal{C}$) are predicted from \mathbf{P}_i^c as in Figure 7. ($\Delta\mathcal{R}, \Delta\mathcal{C}$) are predicted without supervision, which asks for the network to search for the optimal adjustment in spatial alignment (see Equation 1 in our paper).

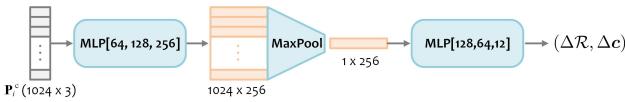


Figure 7: Rotation and translation adjustment.

A.3. Shape Generator

In section 3.3, we design the shape generator with two parts (see Figure 3 of the paper): a shape encoder extended with skip propagation; and a shape decoder based on conditional batch normalization.

Shape Encoder. The layer specification of the denoiser in skip propagation is illustrated in Figure 8. The PointNet

encoder [49] designed with residual connection is shown in Figure 9. It takes the extended point clusters as input (see section 3.3 in the paper) and outputs the new proposal features $\{f_p^* \in \mathbb{R}^{D_s}\}$, $D_s = 512$ to decode shapes.

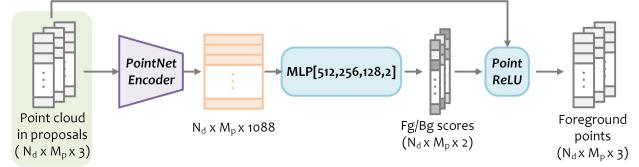


Figure 8: Denoiser in skip propagation.

Shape Decoder. We build the shape decoder as a probabilistic generative model. The latent encoder [41] is fed with the proposal feature $\{f_p^*\}$, spatial points $\{\mathbf{p} \in \mathbb{R}^3\}$ with corresponding occupancy values $\{o\}$. It outputs the mean and standard deviation (μ, σ) to approximate the standard normal distribution. We illustrate the latent encoder in Figure 10, where the 2,048 spatial points are randomly sampled in the shape bounding cube. In our method, we separately sample 1,024 points inside and 1,024 points outside the shape mesh. From the predicted distribution $N(\mu, \sigma)$, we sample a latent code $\mathbf{z} \in \mathbb{R}^L$ ($L = 32$) to predict the occupancy values $\{o\}$ from $\{\mathbf{p}\}$ conditioned on f_p^* for each object. The shape decoder is based on the Conditional Batch Normalization [16, 17] layers, which is illustrated in Figure 11.

During test, we directly initialize the latent codes with zeros. As discussed in section 3.3 of our paper, we uniformly sample spatial points in the object bounding box, and use Marching Cubes [40] to extract the iso-surface as object meshes. Specifically, we adopt the efficient Multi-resolution Iso-Surface Extraction (MISE) algorithm [41] to improve the spatial sampling efficiency and extract meshes under 128-d occupancy grids.

A.4. Weights in Loss Functions

We list the weights to balance different loss functions as follows. We set $\lambda_{cls} = 0.1$ as the hybrid ratio of combining the classification and regression losses, i.e., $\lambda_{cls}\mathcal{L}_{cls} + \mathcal{L}_{reg}$, in defining the scale loss \mathcal{L}_s and heading angle loss \mathcal{L}_θ . Then the box loss \mathcal{L}_{box} in our paper can be denoted as:

$$\mathcal{L}_{box} = \mathcal{L}_v + \mathcal{L}_c + \mathcal{L}_s + \mathcal{L}_\theta + \lambda_l \mathcal{L}_l + \lambda_{obj} \mathcal{L}_{obj}, \quad (4)$$

where $\lambda_{obj} = 0.5$, $\lambda_l = 0.1$. For shape loss in Equation 3 of our paper, we set $\lambda_{seg} = 1e2$. Then the total loss for end-to-end training can be summarized as:

$$\mathcal{L} = \mathcal{L}_{box} + \lambda \mathcal{L}_{shape}, \quad (5)$$

where $\lambda = 5e-3$.

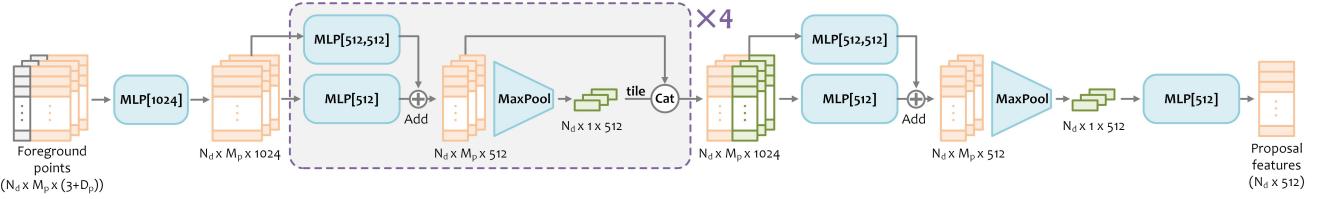


Figure 9: PointNet-based shape encoder with residual connection.

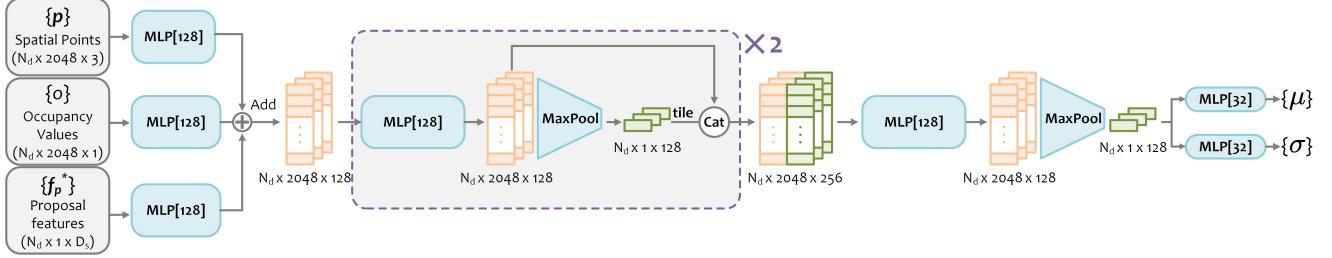


Figure 10: Latent encoder for probabilistic shape generation.

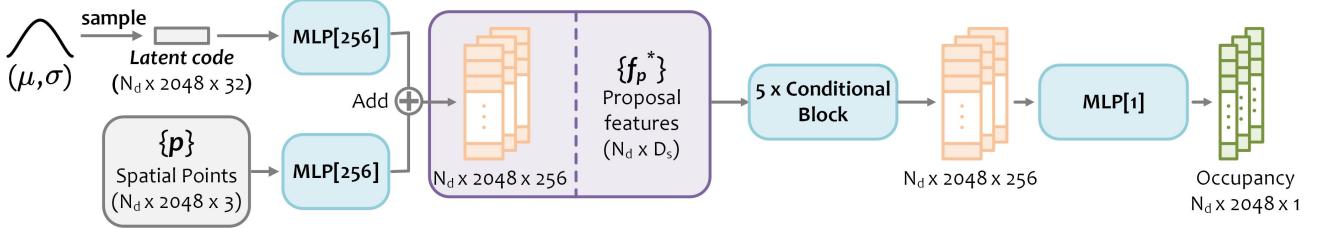


Figure 11: Shape decoder with Conditional Batch Normalization layers.

B. Efficiency and Memory in Inference

We train our network with two NVIDIA TITAN-Xp GPUs and test it on a single GPU. We also compare the inference timing in single forward pass and GPU memory usage with [25] (see Table 8). It shows that our method have comparable efficiency with the state-of-the-art but requires much fewer memory ($\approx 1/3$), which indicates that learning shapes directly from the raw point cloud consumes fewer computation resource and hardware requirement than processing 3D scenes with TSDF grids.

	Max. Time (s)	Max. Memory (MB)
RevealNet [25]	0.72	4273
Ours	0.68	1239

Table 8: Maximal inference time (seconds) and GPU memory (MB) of a single forward pass in ScanNet v2 [11].

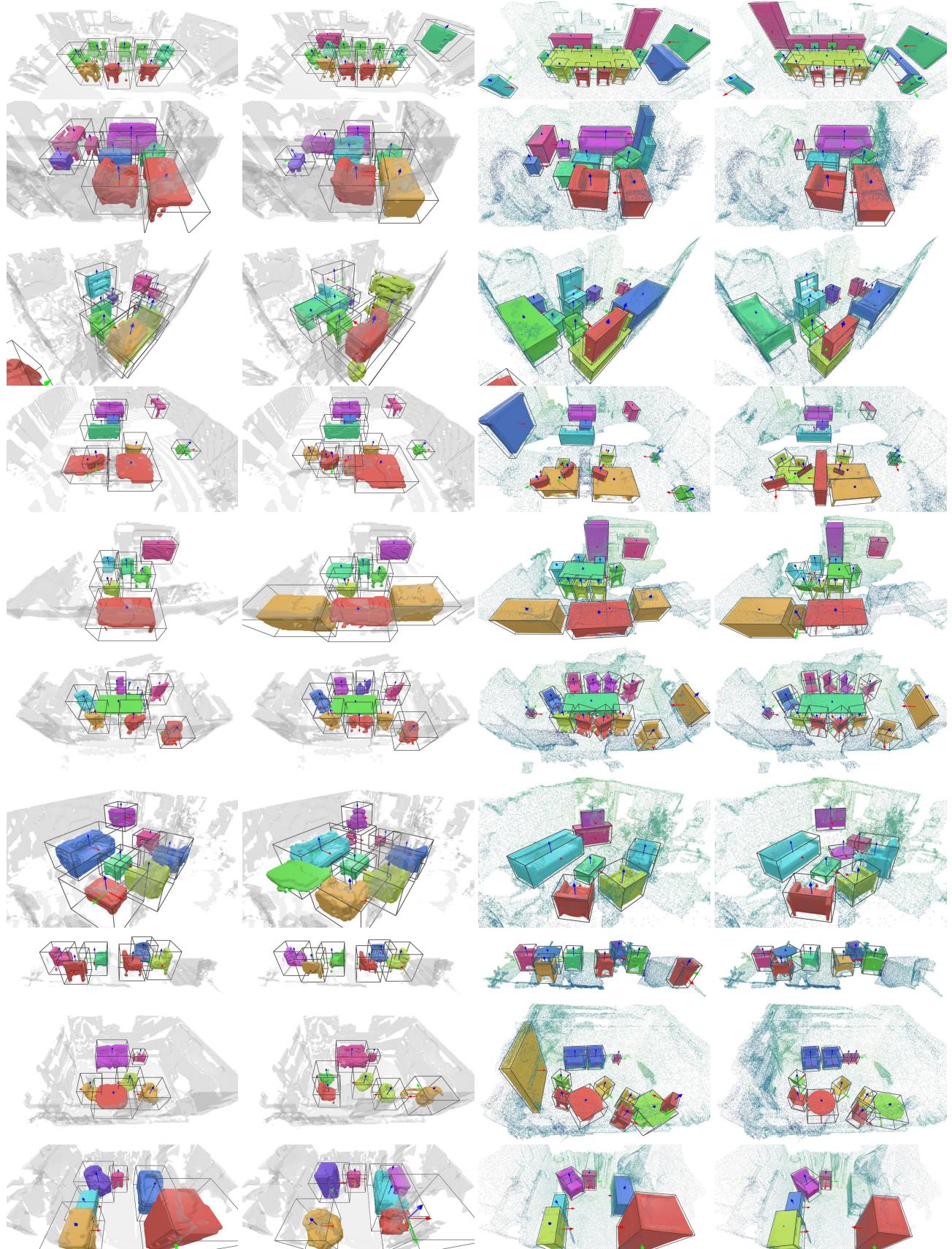
C. Detailed Quantitative Comparisons

In this section, we list the per-category scores of 3D detection and object reconstruction in Table 9 and Table 10

(refer to section 5.2 in the paper).

D. More Qualitative Comparisons

We provide more qualitative comparisons of semantic instance reconstruction with [25] in Figure 12. As in RevealNet [25], we use the groundtruth objects in Scan2CAD [3] for supervision. Note that the groundtruth CAD models in Scan2Cad are only partially labeled.



(a) RevealNet [25] (Geo+Image)

(b) RevealNet [25] (Geo Only)

(c) Ours (Geo Only)

(d) GT

14

Figure 12: Qualitative results of semantic instance reconstruction on ScanNet v2 [11]. Note that RevealNet [25] preprocesses the scanned scenes into TSDF grids, while our method only uses the raw point clouds.

	Input	table	bathtub	trashbin	sofa	chair	cabinet	bookshelf	display	mAP
3D-SIS [24]	Geo+RGB	42.76	3.03	20.04	34.39	63.26	21.54	16.92	3.69	25.70
MLCVNet [63]	Geo Only	44.51	22.39	10.10	53.13	78.74	26.34	22.93	8.90	33.40
RevealNet [25]	Geo Only	35.64	14.94	26.77	29.96	53.18	26.63	15.89	31.30	29.29
Ours (w/o joint)	Geo Only	46.67	19.09	15.30	51.71	77.22	24.62	18.58	7.03	32.63
Ours (w/ joint)	Geo Only	49.90	23.63	15.69	52.34	79.88	26.72	23.20	9.23	35.10

Table 9: 3D object detection on ScanNet v2. 3D-SIS [24] and RevealNet [25] results are provided by the authors. MLCVNet results are retrained with the original network [63]. Scores above are evaluated with mAP@0.5.

	resolution	table	bathtub	trashbin	sofa	chair	cabinet	bookshelf	display	3D IoU
RevealNet [25]	avg. 27-d	17.43	12.64	17.90	28.73	29.61	20.78	18.05	18.68	20.48
Ours (w/o joint)	16-d	16.08	32.74	36.24	46.51	35.53	45.71	33.55	39.63	35.75
Ours (w/o joint)	32-d	11.65	28.07	35.86	39.97	28.89	44.27	23.54	29.44	30.21
Ours (w/o joint)	64-d	20.68	29.46	25.43	24.19	23.55	30.84	22.43	23.20	24.97
Ours (w/ joint)	16-d	19.22	32.55	36.92	46.73	37.05	49.25	35.14	39.28	37.02
Ours (w/ joint)	32-d	15.24	28.55	36.09	41.47	30.91	47.03	24.94	30.27	31.81
Ours (w/ joint)	64-d	22.12	29.24	28.12	27.80	26.05	33.75	23.27	22.87	26.65

Table 10: Comparisons on object reconstruction. Scores above are measured with 3D mesh IoU.