Point Scene Understanding via Disentangled Instance Mesh Reconstruction

Jiaxiang Tang[1], Xiaokang Chen[1], Jingbo Wang[2], and Gang Zeng[1]

[1] Key Laboratory of Perception (MoE), School of AI, Peking University
[2] Chinese University of Hong Kong
{tjx, pkucxk}@pku.edu.cn, wj020@ie.cuhk.edu.hk, zeng@pku.edu.cn

Abstract. Semantic scene reconstruction from point cloud is an essential and challenging task for 3D scene understanding. This task requires not only to recognize each instance in the scene, but also to recover their geometries based on the partial observed point cloud. Existing methods usually attempt to directly predict occupancy values of the complete object based on incomplete point cloud proposals from a detection-based backbone. However, this framework always fails to reconstruct high fidelity mesh due to the obstruction of various detected false positive object proposals and the ambiguity of incomplete point observations for learning occupancy values of complete objects. To circumvent the hurdle, we propose a Disentangled Instance Mesh Reconstruction (DIMR) framework for effective point scene understanding. A segmentation-based backbone is applied to reduce false positive object proposals, which further benefits our exploration on the relationship between recognition and reconstruction. Based on the accurate proposals, we leverage a mesh-aware latent code space to disentangle the processes of shape completion and mesh generation, relieving the ambiguity caused by the incomplete point observations. Furthermore, with access to the CAD model pool at test time, our model can also be used to improve the reconstruction quality by performing mesh retrieval without extra training. We thoroughly evaluate the reconstructed mesh quality with multiple metrics, and demonstrate the superiority of our method on the challenging ScanNet dataset. Code is available at https://github.com/ashawkey/dimr.

Keywords: Point Scene Understanding, Mesh Generation and Retrieval, Point Instance Completion

1 Introduction

Semantic scene reconstruction can facilitate numerous real-world applications, such as robot navigation, AR/VR and interior design. This task aims to understand the semantic information of each object and recover their geometries from partial observations (*e.g.* point cloud from 3D scans). Several previous methods only focus on object recognition in the scene [37, 29, 6, 32, 33, 13, 63] by semantic and instance segmentation, or the completion of the partial observed point cloud [55, 35, 28, 17, 19, 16]. In order to further explore both semantic

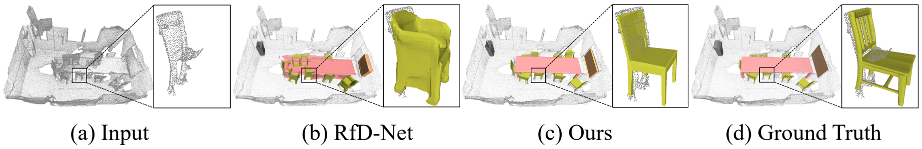


Fig. 1: Point Scene Instance Mesh Reconstruction. With an incomplete point cloud scene as input, our method learns to recognize each object instance and reconstruct a complete mesh that matches the input observation well.

and geometry information, in this paper, we aim to jointly complete these two different tasks in one framework.

Recently, researchers begin to explore the relationship of semantic and geometry information for scene understanding. Semantic Scene Completion [57, 43, 24, 7] reconstructs occluded geometry by performing semantic segmentation for both visible and occluded space in dense voxel grids. Similarly, RevealNet [31] performs instance segmentation in dense voxel grids. Due to the demanding memory requirement, these works are typically limited by the low-resolution dense voxel grids and can not reconstruct high fidelity objects in the scene. RfD-Net [49] first proposes to work directly on sparse point clouds, which can recognize and reconstruct objects in high-resolution mesh representation. However, as shown in Figure 1(b), this Reconstruction-from-Detection pipeline always fails to reconstruct high fidelity objects. In general, the reason can be mainly categorized into two aspects. The first one is the numerous false positive proposals from the detection module. These false positive proposals cause the mismatch between the incomplete point clouds and the complete mesh, thus obstructing training an effective shape completion network. The second one is the structure ambiguity caused by incomplete point observations for directly learning occupancy values of complete objects. Therefore, to further explore this problem, we should answer the following two questions for this task: *i*) Does the accurate foreground object proposals improve the reconstruction quality? *ii*) How to mitigate the structure ambiguity of incomplete point cloud for mesh reconstruction?

In this paper, we propose our Disentangled Instance Mesh Reconstruction framework to answer these two questions. Our pipeline contains two stages, namely instance segmentation and instance mesh reconstruction. Comparing against state-of-the-art point cloud detection approaches [53, 68, 44], we observe that instance segmentation framework [37, 6] can reduce the false positive rate of object proposal significantly. Therefore, we generate the object proposal for object completion based on instance segmentation. With the proposals from the instance segmentation framework, the quality of completed objects is improved consequentially. For the second question, we propose a disentangled instance mesh reconstruction approach to recover high fidelity mesh of each incomplete object. Different from directly learning occupancy values of complete objects based on incomplete point observations [49], we propose to disentangle the shape completion and mesh generation for mesh reconstruction. The shape completion

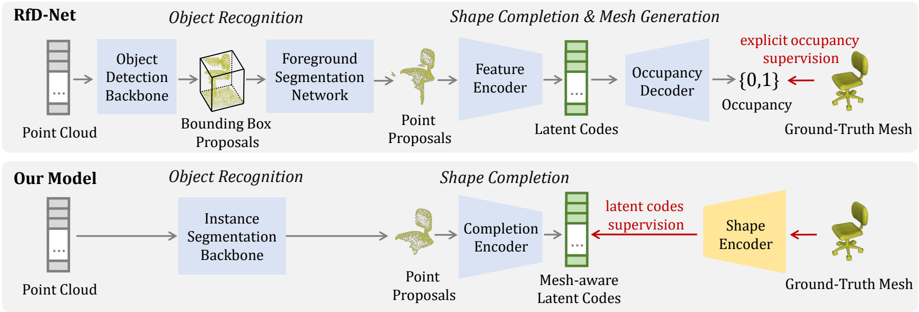


Fig. 2: A comparison of the training process between our method and RfD-Net. For object recognition, RfD-Net uses a detection-based backbone to predict bounding boxes, then extracts the foreground points with a segmentation network, while we apply a straight-forward instance segmentation backbone. Furthermore, we disentangle shape completion and mesh generation by adopting a pre-trained mesh autoencoder and supervise shape completion by latent codes. The mesh generation part is only used during inference (not shown in the figure), mitigating the issue of difficult explicit occupancy supervision in RfD-Net. Without harming the end-to-end training of object recognition and shape completion, our method significantly improves the quality of generated meshes.

module aims to recover the necessary structure information of incomplete objects to mitigate the ambiguity for high-fidelity mesh reconstruction, which is the goal of the mesh generation module. Especially, our shape completion module does not focus on direct completion of mesh, due to the noisy information from input point observations. More effectively, we propose the mesh-aware latent code as the supervision for our shape completion module. The target latent code is encoded from the complete point cloud by a pre-trained encoder, and can be used for mesh reconstruction by a pre-trained decoder, namely the mesh generator. Therefore, the structure information of the complete point cloud for mesh reconstruction is encoded into our mesh-aware latent code, and our shape completion module can learn this structure information directly. After training, the structure information of incomplete point observations can be recovered by this module and mitigate the ambiguity for mesh reconstruction. With the pre-trained mesh generator, our method can generate high-quality meshes consistent with the point observations as shown in Figure 1 (c). Furthermore, if we have access to the CAD model pool, our model can be used to search the nearest neighbors in the latent code space to perform mesh retrieval or assist mesh generation, without the need of extra training.

To summarize, the contributions of this paper are as follows:

- We analyze the weaknesses of the previous detection based framework and propose a new pipeline for point scene instance mesh reconstruction, which first performs instance segmentation on incomplete point scenes and then completes each object instance with a mesh that matches the observed points.

- We design a disentangled instance mesh reconstruction strategy to mitigate the ambiguity of learning complete shapes from incomplete point cloud observations, by leveraging a mesh-aware latent code space. Furthermore, it can also be used for mesh retrieval with the access to a provided model pool.
- We studied multiple metrics to measure the performance in mesh completion quality, and proposed a new metric to measure point-to-mesh mapping quality. Results show that our method performs better than previous state-of-the-arts on the challenging ScanNet dataset, especially on complex structures such as chairs and tables.

2 Related Work

2.1 Point Scene Instance Segmentation

Instance segmentation has been an important topic for point scene understanding with the availability of large-scale point cloud scene datasets. Current methods can be categorized into detection-based and segmentation-based methods. Detection-based methods [20, 30, 70] first regress 3D bounding boxes and then mask out background points inside each box to get the final instance segmentation. However, the two-step pipeline is not straightforward and usually inefficient. Instead, segmentation-based methods [37, 64, 52, 29, 40, 26, 6, 42, 73] directly predict semantic segmentation and then cluster points into instance proposals. For example, PointGroup [37] uses sparse 3D CNNs [23, 69, 45, 59] to extract point cloud features, and propose a dual-set clustering algorithm to better distinguish the void space between object instances. Later works [6, 29] mainly focuses on more efficient and concise instance clustering algorithms such as dynamic convolution and hierarchical aggregation. We choose the segmentation-based backbone for its simplicity, and bridge instance segmentation to mesh reconstruction in end-to-end training.

2.2 3D Shape Completion

Object Completion. This line of research mainly focuses on shape completion of single objects. Many works [65, 61, 50, 71] focus on the completion of point cloud shapes, with incomplete point clouds as the input and completed point clouds as the output. However, these methods usually complete up to a limited number of points which is not enough to represent high resolution shapes due to the sampling problem. Other works choose dense voxel grids [58, 18, 27] or implicit functions [12, 11] to perform shape completion. Many works [11, 47, 41, 67, 51] adopt an autoencoder architecture to learn a compact latent code for each shape. BSP-Net [9, 10] proposes to approximate shapes with a Binary Space Partitioning (BSP) tree, which shows good results on mesh reconstruction from dense voxel grids and single view images.

Scene Completion. Instead of focusing on single objects, scene completion aims to complete all objects from a partial observation such as a 3D scan. Early works usually start from volumetric representations and the completion task can be viewed as a dense labeling task on voxel grids. Semantic Scene Completion [57, 24, 43, 7, 17, 8, 60] voxelizes the point cloud into dense voxel grids and predicts semantic labels of all voxels in both visible and occluded regions. RevealNet [31] proposes semantic instance completion, which performs object detection on voxel grids and then completes each instance within the cropped voxel grids. Other works focus on mesh representations. Total 3D understanding [48, 72, 21] performs object detection and mesh reconstruction on RGB images. RfD-Net [49] first performs semantic instance completion on point clouds directly and generates completed instance meshes. It adopts a detection-based backbone and uses implicit functions for mesh reconstruction, demonstrating that these two tasks are complementary. Assuming the availability of a CAD model pool, CAD retrieval aims to find the best-fitting CAD models and align them to the point scenes [2, 3, 14, 25, 22], images [39], or videos [46], optionally allowing for deformation of single objects [62, 36]. Our method follows this line of research, with point clouds as the direct input and reconstructed instance meshes as the output. Differently, we separate shape completion and mesh generation tasks to ease the training process with the proposed latent instance mesh reconstruction.

3 Method

We introduce our pipeline as illustrated in Figure 3. Overall, the pipeline consists of two training stages: point-wise learning and proposal-wise learning. The first stage (Section 3.1) takes a sparse 3D CNN backbone to perform point-wise predictions including semantic labels, instance center offsets, and rotation angles. In the second stage (Section 3.2), another sparse 3D CNN is used to predict proposal-wise results including residual bounding boxes, confidence scores, and the latent distributions of complete meshes. Only in inference, the latent codes sampled from these distributions are decoded to generate compact meshes (Section 3.3), which are transformed back to the world coordinate system with the refined bounding boxes to compose the final reconstructed scene.

3.1 Learning Point-wise Features

In this stage, we focus on learning point-wise features including the semantic labels for semantic segmentation, offsets from the instance center for instance segmentation, and instance rotation angles for bounding box regression. We follow [37] and use a sparse 3D U-Net [23, 54] as the backbone to extract features. The input to the network is a point set $\mathbb{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$, where each point is described by its coordinate $\mathbf{p}_i = (x_i, y_i, z_i), i \in [1, N]$. These points are voxelized before being fed to the backbone. To obtain the per-point feature, we map the voxel feature from the backbone back to the point and get

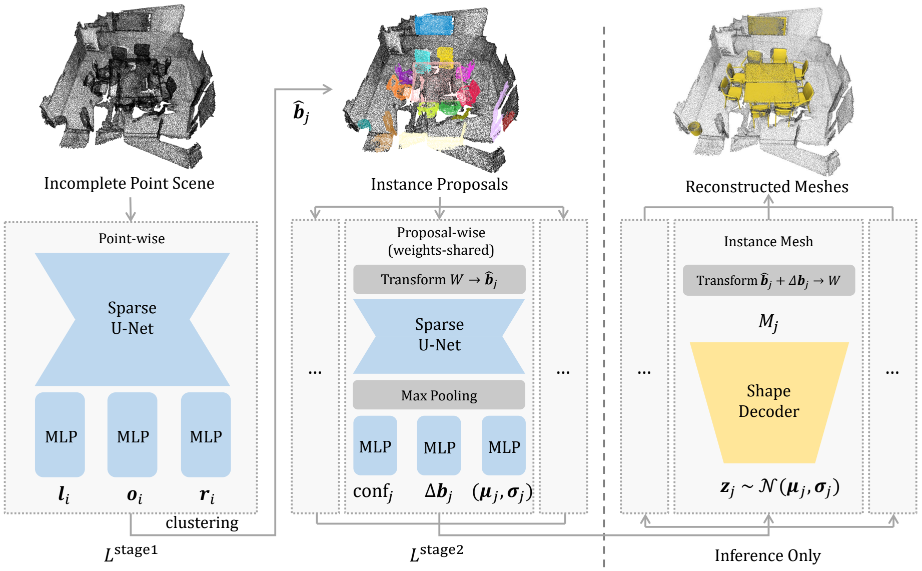


Fig. 3: Overview of the network architecture. The network first learns point-wise features including semantic labels $\mathbf{l}_i$, instance offsets $\mathbf{o}_i$ and instance rotations $\mathbf{r}_i$. Then, instance proposals are clustered and fed to the second stage for learning proposal-wise features including confidence scores conf_j, residual bounding boxes $\Delta\mathbf{b}_j$, and latent shape distributions $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)$. During inference, instance meshes are generated by decoding the mesh-aware latent codes sampled from the latent shape distributions.

$\mathbf{F}_{\text{point}} \in \mathbb{R}^{N \times D_{\text{point}}}$, where D_{point} is the feature dimension. Then, three Multi-Layer Perceptrons (MLPs) are applied to regress three point-wise targets respectively. For the semantic label, the prediction is the classification logits $\mathbf{l}_i \in \mathbb{R}^C$, where C is the total number of classes. For the instance offset, the prediction is the offset $\mathbf{o}_i = (\Delta x_i, \Delta y_i, \Delta z_i)$ from the current point to the instance center it belongs to. In addition to these two regular heads for instance segmentation [37, 6], we use a third head to predict the orientation of the instance that covers the current point to build an approximate oriented bounding box. We only predict the rotation angle $r_i \in [-\pi, \pi)$ along the z-axis following [49], since the rotation along x, y-axes for most instances can be ignored.

To optimize the afore-mentioned objectives, we use the cross-entropy loss $L_{\text{cls}}^{\text{semantic}}$ for semantic segmentation, the L1 Loss $L_{\text{reg}}^{\text{offset}}$ for instance offset regression, and follow [49, 34] to disentangle the angle loss into a hybrid of classification and regression loss $\mathcal{L}_{\text{cls}}^{\text{angle}} + \mathcal{L}_{\text{reg}}^{\text{angle}}$. So far, the loss function for the first stage $\mathcal{L}^{\text{stage1}}$ can be concluded as the sum of these four parts.

3.2 Learning Proposal-wise Features

The second stage handles proposal-wise predictions that bridge instance segmentation to instance mesh reconstruction. Given the point-wise predictions from stage one, we first apply a clustering algorithm [37] to group the whole scene's point cloud into L instance point cloud proposals $\mathcal{P} = \{P_1, P_2, \cdots, P_L\}$. We then transform each instance point cloud to its canonical coordinate system for better proposal-wise feature learning [56]. Specifically, each instance point cloud $P_j \in \mathcal{P}$ is: 1) recentered at the mean instance center $\bar{\mathbf{c}}_j = \frac{1}{|P_j|}\sum_{i \in P_j}(\mathbf{p}_i + \mathbf{o}_i)$; 2) rotated along z-axis for the negative mean rotation angle $-\bar{r}_j = -\frac{1}{|P_j|}\sum_{i \in P_j} r_i$ to make the instance front-facing; and 3) scaled into $[0, 1]$ on each axis by dividing $\mathbf{s}_j$, where $\mathbf{s}_j \in \mathbb{R}^3$ is the approximate instance scale calculated from the minimum and maximum coordinates of the rotated points. Another voxelization is applied on each instance proposal to extract proposal-wise features. The transformed instance points with their features $\mathbf{F}_{\text{point}}$ are fed into the second sparse 3D U-Net, after which a max-pooling layer is used to output the proposal-wise features $\mathbf{F}_{\text{prop}} \in \mathbb{R}^{L \times D_{\text{prop}}}$, where D_{prop} is the feature dimension. This allows the point-wise features learned in stage one to be smoothly propagated to later modules. The original scale information $\mathbf{s}_j$ is preserved by being concatenated to the features.

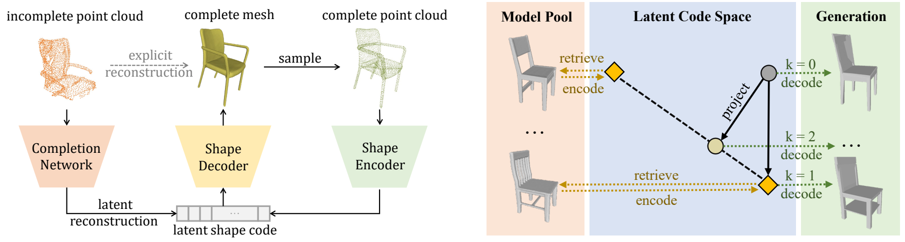
To reconstruct instance meshes from $\mathbf{F}_{\text{prop}}$, we still need to regress three targets: proposal confidence, residual bounding box and latent shape distributions.

Proposal confidence. An MLP followed by a sigmoid function is applied to regress the confidence value $\text{conf}_j \in [0, 1]$ for proposal P_j. The ground truth for the confidence is decided by the largest point Intersection over Union (IoU) between the proposal and ground-truth instances following [37].

Residual bounding Box. From the first stage, we already have an initial 7 Degree-of-Freedom (DoF) oriented bounding box $\hat{\mathbf{b}}_j = \{\bar{r}_j, \bar{\mathbf{c}}_j, \mathbf{s}_j\}$. However, this bounding box is inaccurate due to partial observation and occlusion, especially for the instance scale (*e.g.*, missing of chair legs leads to underestimated scale on the z-axis). We therefore use an MLP to predict a residual bounding box $\Delta\mathbf{b}_j$ to refine this initial bounding box, and the final bounding box is given by $\mathbf{b}_j = \hat{\mathbf{b}}_j + \Delta\mathbf{b}_j$.

Latent shape distributions. To address the ambiguity problem in shape completion, a probabilistic generative model is usually adopted [66, 1, 49]. We take a similar way by assuming that the complete shape is sampled from a latent Gaussian distribution, and learn it through the reparameterization trick [38]. An MLP is used to regress the mean and standard deviation $\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j \in \mathbb{R}^{D_{\text{shape}}}$, where D_{shape} is the latent shape code dimension. To allow supervised learning, we need to know the ground-truth latent distribution $(\boldsymbol{\mu}_j^{\text{gt}}, \boldsymbol{\sigma}_j^{\text{gt}})$ of ground-truth meshes, which will be described in Section 3.3.

The loss function for the second stage $\mathcal{L}^{\text{stage2}}$ further adds three regression terms on the basis of $\mathcal{L}^{\text{stage1}}$, *i.e.*, the confidence loss $\mathcal{L}_{\text{reg}}^{\text{conf}}$, the bounding box loss $\mathcal{L}_{\text{reg}}^{\text{bbox}}$, and the latent distribution loss $\mathcal{L}_{\text{reg}}^{\text{latent}}$, all using the weighted smooth L1 loss to alleviate the class imbalance in object proposals.



(a) Disentangled Instance Mesh Reconstruction (b) Mesh Retrieval and Generation

Fig. 4: (a) Instead of explicitly predicting a complete mesh in 3D space, we leverage a mesh-aware latent code space to disentangle shape completion and mesh generation. (b) The gray circle is the predicted latent code from the observed partial point cloud. We can directly use it to generate a mesh via the decoder (green right arrow with $k = 0$). With access to the CAD meshes or their latent codes (yellow diamonds) from the model pool, we can further perform model retrieval (yellow left arrows) or assisted generation by projecting the latent code to the nearest k neighbors (green right arrows with $k > 0$).

3.3 Disentangled Instance Mesh Reconstruction

Instance mesh reconstruction from a partially observed point cloud is a challenging task, requiring both shape completion and mesh generation. Previous methods perform these two tasks as a whole, but fail to generate high fidelity meshes that match the observation. The problems are two-fold: 1) The difficulty of optimizing a conditioned mesh generator with the detection network, where lots of false positive proposals are used as inputs. 2) The ambiguity of learning complete shapes from incomplete point cloud observations. To handle these problems, we propose a disentangled mesh reconstruction approach as illustrated in Figure 4a. The core idea here is to disentangle shape completion and mesh generation into two stages. First, we pre-train a mesh Variational Autoencoder (VAE) to encode 3D meshes into a latent code space, which can be viewed as a mesh generator. Note that this mesh generator is trained with complete GT meshes, and there is no ambiguity in shape learning since no completion happens here. For shape completion, we can simply train another encoder that maps incomplete instance proposals into the same latent code space. This completion encoder is supervised with these low-dimension latent codes as described in Section 3.2, which are much easier to optimize compared to high-dimension conditioned occupancy values. Thus, the mesh generation part is detached from the training process of object recognition and shape completion.

We adopt BSP-Net [9, 10], which proposes an efficient approach to approximate low-poly meshes by learning convex decomposition, as the autoencoder model. Specifically, we adopt a Conditional VAE (CVAE) variant for better generation quality. Following [9], we sample complete point clouds from the CAD models and voxelize them as the input to the encoder, which outputs a latent distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ that characterizes the shape. Then, we sample a latent code $\mathbf{z}$ from this distribution, and use the decoder to output a set of planes with the convex decomposition to generate the polygonal meshes. In addition to the original BSP loss $\mathcal{L}^{\text{bsp}}$ in [9], a KL loss $\mathcal{L}^{\text{KL}}$ weighted by 0.1 is added as regularization. More details can be found in the supplementary material.

After convergence, we use the encoder to extract the ground-truth latent distribution for each ground-truth mesh M_j^{gt}:

$$\boldsymbol{\mu}_j^{\text{gt}}, \boldsymbol{\sigma}_j^{\text{gt}} = \text{Enc}(M_j^{\text{gt}}) \tag{1}$$

Therefore, to reconstruct mesh M_j from a partially observed point instance, we only need to regress the latent distribution and sample a latent code $\mathbf{z}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)$, then decode it through:

$$M_j = \text{Dec}(\mathbf{z}_j) \tag{2}$$

Optional post-processing like the Iterative Closest Point (ICP) algorithm can be used to further fine-tune the mesh location. By default, we use the expectation as the latent code so $\mathbf{z}_j = \boldsymbol{\mu}_j$, but we can also sample different latent codes for different explanations of the partial point observation to address the ambiguity problem. Furthermore, if we have access to the model pool $\mathcal{M}$ at test time, our model can also perform CAD retrieval task without any further training, by searching the nearest mesh from the model pool in the latent space:

$$M_j^{\text{retr}} = \arg\min_{m \in \mathcal{M}} ||\mathbf{z}_m - \mathbf{z}_j||_2 \tag{3}$$

where $\mathbf{z}_m$ is the latent code of CAD model m. However, maintaining the whole model pool requires extra storage (about 238MB for the 2238 models used in Scan2CAD). Another option is to only maintain the latent codes $\{\mathbf{z}_m | m \in \mathcal{M}\}$, which is a 256-d vector for each mesh (about 2.2MB for the same models). These latent codes can serve as priors to assist the mesh generation, by projecting the predicted latent code to the hyperplane spanned by the nearest k latent codes $\{\mathbf{z}_{n1}, \mathbf{z}_{n2}, \cdots, \mathbf{z}_{nk}\}$ from the model pool:

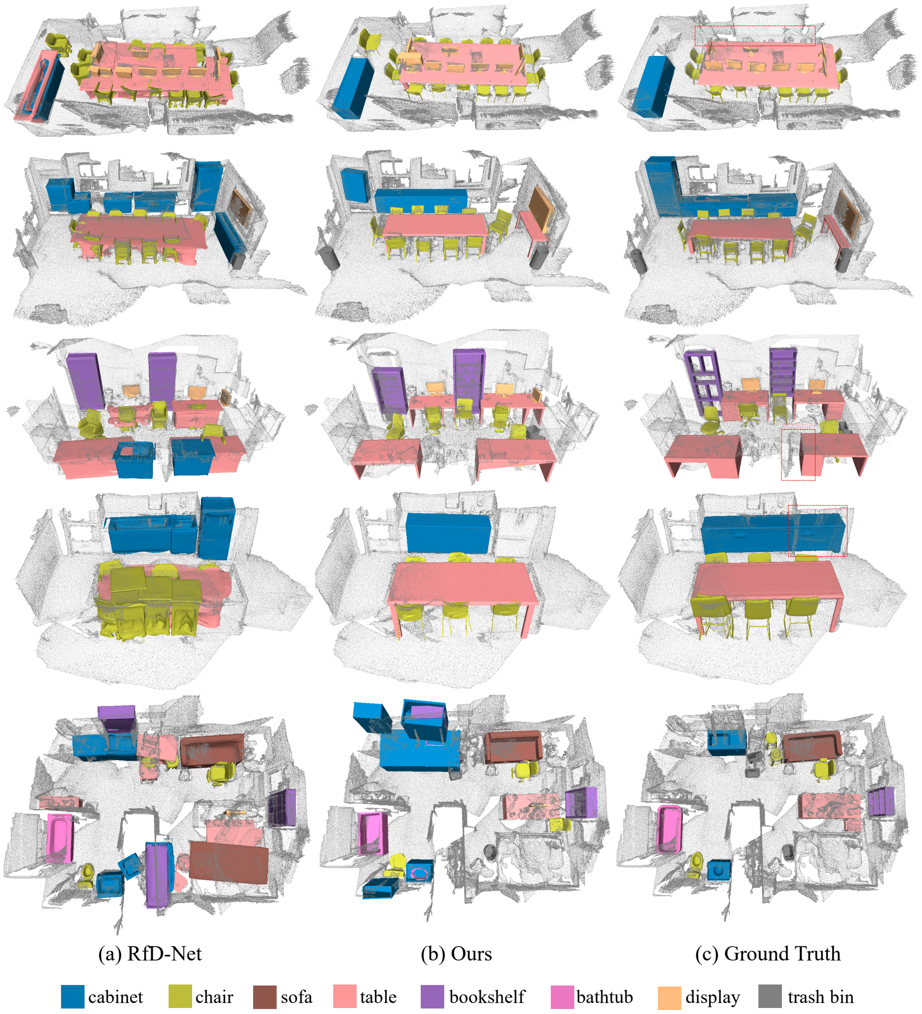
$$M_j^{\text{proj}} = \text{Dec}(\text{proj}_{\text{span}\{\mathbf{z}_{n1}, \mathbf{z}_{n2}, \cdots, \mathbf{z}_{nk}\}}(\mathbf{z}_j)) \tag{4}$$

Figure 4b illustrates the relationship between these methods.

4 Experiment

4.1 Experimental Settings

Datasets. Three datasets are involved in our experiments: ScanNet v2 [15], ShapeNet [4], and Scan2CAD [2]. The ScanNet dataset consists of 1,513 real world indoor scene scans. Inline with [49, 31], The official data split is used in all experiments. Only the incomplete point clouds are used as the input data, with the semantic and instance labels as the point-level supervision. The Scan2CAD



(a) RfD-Net

(b) Ours

(c) Ground Truth

Fig. 5: Qualitative Comparison on the ScanNet dataset. Please zoom in to see details. We only visualize meshes with confidence larger than 0.5 for RfD-Net as in [49], and 0.3 for Ours. Red-dash boxes show missing or incorrect human annotations in the ground truth, *e.g.*, missing chairs, incorrectly scaled table and cabinet. More visualizations can be found in the supplementary material.

dataset provides the alignment of CAD models in ShapeNet to scenes in ScanNet. We use the aligned CAD models as the proposal-level supervision.

It's worth noting that the semantic and instance labels of ScanNet point clouds are inconsistent with Scan2CAD meshes. The point cloud instance segmentation literature [37, 29, 6] usually adopts a 20-class label system, with 2 stuff

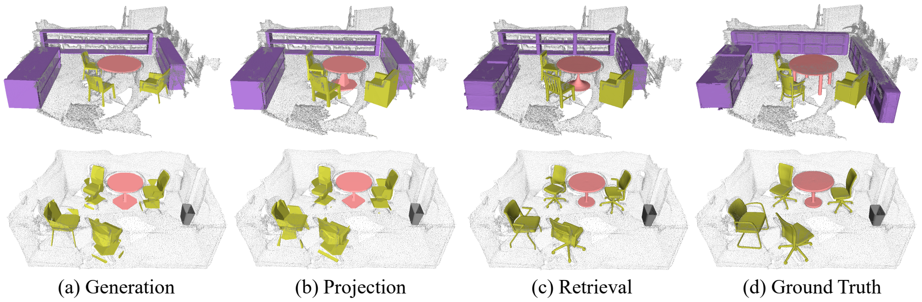


Fig. 6: Comparison between mesh generation, assisted generation (projection) and retrieval mode of our method.

categories and 18 object categories. However, the instance mesh reconstruction literature [49] only uses 8 object categories. To bridge between these two tasks, we make a compatible label system to both tasks and relabel the instance segmentation ground truths based on the new label system. Note that we don't introduce any new manual labeling, but simply rearrange the two sources of label information. The details of the label mapping and relabeling can be found in the supplementary material.

Evaluation Metrics. The quality of point scene instance mesh reconstruction could be evaluated in two aspects: the completion quality and the mapping quality. The completion quality measures how well the reconstructed meshes match the human-annotated ground-truth meshes, and the mapping quality measures how well the reconstructed mesh surfaces match the observed point clouds.

Completion quality. Several approaches have been proposed to measure the similarity between different meshes. For voxel-based methods, [49] voxelizes the meshes with a fixed voxel resolution and calculates the 3D Intersection over Union (IoU) of the prediction and the ground truth. For point-based methods, [11, 9] sample point clouds with a fixed number from the mesh surfaces and calculate the Chamfer Distance (CD). For mesh-based methods, [9, 5] choose to render multi-view 2D images and calculate the Light Field Distance (LFD). All of these metrics can only partially represent the underlying mesh similarity. For example, 3D IoU requires the predicted mesh to occupy the exact voxels as the ground-truth mesh, CD measures the distance between mesh surfaces, and LFD compares more on visual appearance. Previous works [49, 31] only adopt 3D IoU as the metric, but we argue that 3D IoU at a coarse voxel resolution fails to reflect the quality of mesh surface and visual appearance, which are also important for measuring mesh similarity. For a thorough comparison, we adopt all the three metrics with different thresholds to determine whether a predicted mesh can match a ground-truth mesh, and report the 3D detection mean AP over all classes.

Mapping quality. As shown in Figure 1, the human-annotated ground-truth meshes may not be the only plausible reconstruction due to the ambiguity of input. Therefore, only using the completion quality may lead to biased evaluation. To alleviate this problem, we use mapping quality to measure how well the reconstructed mesh surfaces match the observed point clouds. Specifically, we propose the Point Coverage Ratio (PCR), which computes the nearest distance from each observed instance point to the corresponding mesh surface, and uses a threshold to determine whether this point belongs to the reconstructed surface:

$$\mathrm{PCR} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \mathbb{1}_{\{\mathrm{dist}(\mathbf{p}, \mathcal{M}) < \omega\}} \tag{5}$$

where $\mathbf{p}$ is a point from the observed ground-truth instance point cloud $\mathcal{P}$, $\mathbb{1}$ means the indicator function, $\mathcal{M}$ is the reconstructed mesh, $\mathrm{dist}(\mathbf{p}, \mathcal{M})$ is the Euclidean distance from the point to the mesh surface, and ω is the distance threshold. A larger PCR means more observed points are located near the mesh surface, and thus better mapping quality. Similarly, we report the 3D detection mean AP over all classes using PCR as the metric.

Baselines. We mainly compare our results with the current state-of-the-art RfD-Net [49], which is the first work that generates high-resolution mesh for point scene instance reconstruction. The officially released model is used to generate results for evaluation and comparison. Also, we investigate different variants of our method including direct mesh generation, assisted mesh generation by latent code projection and mesh retrieval with different levels of access to the CAD model pool. All methods in our experiments are trained on the same dataset split, and evaluated with the same hyper-parameters.

Implementation Details. We set the voxel size as 0.02m for point-wise sparse U-Net, and 0.05m for proposal-wise sparse U-Net. We use $D_{\text{point}} = 32$, $D_{\text{prop}} = 64$, and $D_{\text{shape}} = 128$. We train 256 epochs for the first stage, and 256 epochs for the second stage with a batch size of 8 on a single Nvidia Tesla V100 GPU. The Adam optimizer is used with an initial learning rate of 0.001 for the first stage and 0.0001 for the second stage. In training, we set the clustering radius as 0.03m following PointGroup [37]. The weight for each loss term defaults to 1.0. In evaluation, we use a multi-scale clustering method at $\{0.01, 0.03, 0.05\}$m to spot more proposals. The nearest neighbour count k is set to 1 for the projection model. The voxel size for 3D IoU calculation and the threshold ω for PCR calculation are both set to 0.047m following [49]. The proposal confidence threshold is set to 0.09, and each proposal should have at least 100 points.

4.2 Comparisons

Quantitative Comparisons. Table 1 shows the quantitative comparisons of completion quality. Our method outperforms state-of-the-art on five out of six evaluation settings in terms of mAP. This benefits from the proposed pipeline that reduces false positive proposals (Table 3) and enhances the quality of generated meshes through the disentangled mesh reconstruction approach. A reason for the lower 3D IoU with 0.5 as the threshold might be that 3D IoU with a

Table 1: **Comparisons on mesh completion quality.** We report mean AP for different metric@threshold. For IoU, higher threshold is more difficult. For CD and LFD, smaller thresholds are more difficult. Better results are in bold. We don't compare with projection and retrieval models since they use extra information.

	IoU@0.25	IoU@0.5	CD@0.1	CD@0.047	LFD@5000	LFD@2500
RfD-Net [49]	42.52	**14.35**	46.37	19.09	28.59	7.80
Ours	**46.34**	12.54	**52.39**	**25.71**	**29.47**	**8.55**
Ours + proj.	46.50	12.59	52.06	24.84	29.95	9.67
Ours + retr.	47.20	12.83	51.77	25.10	30.80	10.12

Table 2: **Comparisons on point-to-mesh mapping quality.** The AP scores are measured with PCR@0.5.

	table	chair	bookshelf	sofa	trash bin	cabinet	display	bathtub	mean
Scan2CAD [2]	36.60	69.31	**65.03**	28.92	56.93	**41.82**	70.81	45.07	45.07
RfD-Net [49]	32.54	76.54	30.66	22.91	40.54	24.37	67.64	52.69	43.49
Ours	**49.78**	**78.64**	29.25	**60.33**	**65.30**	18.75	**76.56**	**75.51**	**56.76**
Ours + proj.	60.57	78.88	28.93	61.00	65.61	18.45	78.02	72.79	58.03
Ours + retr.	62.28	75.57	45.23	52.60	65.27	17.14	76.61	73.81	58.82

Table 3: **Object Recognition Precision.** We report the precision of object recognition at different IoU thresholds. Our method greatly reduces the number of false positive proposals.

	Prec.@0.25	Prec.@0.5
RfD-Net [49]	22.99	7.92
Ours	**43.70**	**15.09**

high threshold discourages meshes with thin structures, since a small displacement can result in huge drop in the metric value, even if the shape is of good quality (*e.g.* the chair in Figure 1). With the access of an external model pool, the projection and retrieval models (denoted as Ours + proj. and Ours + retr.) produce more robust meshes. In particular, the retrieval model achieves a better LFD score, since the retrieved meshes are guaranteed to be rational.

Table 2 shows the quantitative comparisons of mapping quality. We achieve significant improvement in mAP and surpass the human-annotated Scan2CAD dataset. This is to be expected. Since the ShapeNet dataset is synthetic and has a finite number of models, the human-annotated CAD models may not perfectly match the point cloud observations. Interestingly, the results also partially reveal the capability of the CAD model pool. For example, retrieved meshes show better performance on bookshelf, but perform worse on sofa, which means there may

Table 4: **Ablation study of proposed modules.**

ResBox	MSC	ICP	IoU@0.25	CD@0.1	LFD@5000
	✓	✓	41.76	46.51	27.55
✓		✓	43.57	48.26	28.91
✓	✓		42.33	52.19	29.11
✓	✓	✓	**46.34**	**52.39**	**29.47**

be fewer suitable sofa CAD models that fit real world ScanNet data. In such cases, generated meshes can be potentially better.

Qualitative Comparisons. Figure 5 shows the qualitative comparisons. The meshes generated by our method have better visual appearance and more accurate locations. Besides, we show that when the human-annotated ground truths conflict with the observed point clouds, our model can still successfully detect these instances and output plausible meshes. In Figure 6, we also show the results of the projection and retrieval models. With extra information from the model pool, the model can produce more robust meshes.

4.3 Ablation Study

We conducted ablation studies to verify the influence of proposed modules in Table 4. 'ResBox' means we learn a residual bounding box $\Delta \mathbf{b}_j$ to refine the empirical bounding box deduced from observed point clouds. 'MSC' means we use multiple clustering radii to find more proposals at test time. 'ICP' means we apply the ICP algorithm to post-process the reconstructed meshes. The results indicate that the combination of these three modules achieves overall the best performance. In particular, the proposed residual bounding box learning refines the object location and improves all metrics, while ICP post-processing mainly affects the IoU metric.

5 Conclusion

In this paper, we introduce a Disentangled Instance Mesh Reconstruction pipeline for point scene understanding. Our method first performs instance segmentation to generate accurate object proposals, then applies a disentangled instance mesh reconstruction strategy to mitigate the ambiguity of learning complete shapes from incomplete point observations. We evaluate the experimental results on the challenging ScanNet dataset from the perspectives of completion quality and mapping quality, and demonstrate the superior performance of our method.

Acknowledgements. This work is supported by the National Key Research and Development Program of China (2020YFB1708002), National Natural Science Foundation of China (61632003, 61375022, 61403005), Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEK-SenseTime Joint Laboratory of Machine Vision.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: ICML. pp. 40–49. PMLR (2018) 7
2. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M.: Scan2cad: Learning cad model alignment in rgb-d scans. In: CVPR. pp. 2614–2623 (2019) 5, 9, 13
3. Avetisyan, A., Khanova, T., Choy, C., Dash, D., Dai, A., Nießner, M.: Scenecad: Predicting object alignments and layouts in rgb-d scans. arXiv preprint arXiv:2003.12622 (2020) 5
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015) 9
5. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3d model retrieval. In: Computer graphics forum. vol. 22, pp. 223–232. Wiley Online Library (2003) 11
6. Chen, S., Fang, J., Zhang, Q., Liu, W., Wang, X.: Hierarchical aggregation for 3d instance segmentation. arXiv preprint arXiv:2108.02350 (2021) 1, 2, 4, 6, 10
7. Chen, X., Lin, K.Y., Qian, C., Zeng, G., Li, H.: 3d sketch-aware semantic scene completion via semi-supervised structure prior. In: CVPR. pp. 4193–4202 (2020) 2, 5
8. Chen, X., Xing, Y., Zeng, G.: Real-time semantic scene completion via feature aggregation and conditioned prediction. In: ICIP. pp. 2830–2834. IEEE (2020) 5
9. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. In: CVPR (2020) 4, 8, 9, 11
10. Chen, Z., Tagliasacchi, A., Zhang, H.: Learning mesh representations via binary space partitioning tree networks. TPAMI pp. 1–1 (2021). https://doi.org/10.1109/TPAMI.2021.3093440 4, 8
11. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: CVPR. pp. 5939–5948 (2019) 4, 11
12. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: CVPR. pp. 6970–6981 (2020) 4
13. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: CVPR. pp. 3075–3084 (2019) 1
14. Dahnert, M., Dai, A., Guibas, L.J., Nießner, M.: Joint embedding of 3d scan and cad objects. In: ICCV. pp. 8749–8758 (2019) 5
15. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. pp. 5828–5839 (2017) 9
16. Dai, A., Diller, C., Nießner, M.: Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans. In: CVPR. pp. 849–858 (2020) 1
17. Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M.: Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In: CVPR. pp. 4578–4587 (2018) 1, 5
18. Dai, A., Ruizhongtai Qi, C., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: CVPR. pp. 5868–5877 (2017) 4
19. Dai, A., Siddiqui, Y., Thies, J., Valentin, J., Nießner, M.: Spsg: Self-supervised photometric scene generation from rgb-d scans. In: CVPR. pp. 1747–1756 (2021) 1

20. Engelmann, F., Bokeloh, M., Fathi, A., Leibe, B., Nießner, M.: 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In: CVPR. pp. 9031–9040 (2020) 4
21. Engelmann, F., Rematas, K., Leibe, B., Ferrari, V.: From points to multi-object 3d reconstruction. In: CVPR. pp. 4588–4597 (2021) 5
22. Grabner, A., Roth, P.M., Lepetit, V.: 3d pose estimation and 3d model retrieval for objects in the wild. In: CVPR. pp. 3022–3031 (2018) 5
23. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017) 4, 5
24. Guo, Y.X., Tong, X.: View-volume network for semantic scene completion from a single depth image. ArXiv **abs/1806.05361** (2018) 2, 5
25. Hampali, S., Stekovic, S., Sarkar, S.D., Kumar, C.S., Fraundorfer, F., Lepetit, V.: Monte carlo scene search for 3d scene understanding. In: CVPR. pp. 13804–13813 (2021) 5
26. Han, L., Zheng, T., Xu, L., Fang, L.: Occuseg: Occupancy-aware 3d instance segmentation. In: CVPR. pp. 2940–2949 (2020) 4
27. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: ICCV. pp. 85–93 (2017) 4
28. Han, X., Zhang, Z., Du, D., Yang, M., Yu, J., Pan, P., Yang, X., Liu, L., Xiong, Z., Cui, S.: Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. In: CVPR. pp. 234–243 (2019) 1
29. He, T., Shen, C., van den Hengel, A.: Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In: CVPR. pp. 354–363 (2021) 1, 4, 10
30. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: CVPR. pp. 4421–4430 (2019) 4
31. Hou, J., Dai, A., Nießner, M.: Revealnet: Seeing behind objects in rgb-d scans. In: CVPR. pp. 2098–2107 (2020) 2, 5, 9, 11
32. Hu, W., Zhao, H., Jiang, L., Jia, J., Wong, T.T.: Bidirectional projection network for cross dimension scene understanding. In: CVPR. pp. 14373–14382 (2021) 1
33. Hu, Z., Bai, X., Shang, J., Zhang, R., Dong, J., Wang, X., Sun, G., Fu, H., Tai, C.L.: Vmnet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. In: ICCV. pp. 15488–15498 (2021) 1
34. Huang, S., Qi, S., Xiao, Y., Zhu, Y., Wu, Y.N., Zhu, S.C.: Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In: NeurIPS. pp. 207–218 (2018) 6
35. Huang, Y.K., Wu, T.H., Liu, Y.C., Hsu, W.H.: Indoor depth completion with boundary consistency and self-attention. In: ICCV Workshops. pp. 0–0 (2019) 1
36. Ishimtsev, V., Bokhovkin, A., Artemov, A., Ignatyev, S., Niessner, M., Zorin, D., Burnaev, E.: Cad-deform: Deformable fitting of cad models to 3d scans. arXiv preprint arXiv:2007.11965 (2020) 5
37. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: CVPR. pp. 4867–4876 (2020) 1, 2, 4, 5, 6, 7, 10, 12
38. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 7
39. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Mask2cad: 3d shape prediction by learning to segment and retrieve. In: ECCV. pp. 260–277. Springer (2020) 5
40. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3d instance segmentation via multi-task metric learning. In: ICCV. pp. 9256–9266 (2019) 4

41. Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L.: Grass: Generative recursive autoencoders for shape structures. TOG **36**(4), 1–14 (2017) 4
42. Liang, Z., Li, Z., Xu, S., Tan, M., Jia, K.: Instance segmentation in 3d scenes using semantic superpoint tree networks. arXiv preprint arXiv:2108.07478 (2021) 4
43. Liu, S., Hu, Y., Zeng, Y., Tang, Q., Jin, B., Han, Y., Li, X.: See and think: Disentangling semantic scene completion. In: NeurIPS. pp. 261–272 (2018) 2, 5
44. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3d object detection via transformers. In: ICCV. pp. 2949–2958 (2021) 2
45. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. In: NeurIPS. vol. 32, pp. 965–975 (2019) 4
46. Maninis, K.K., Popov, S., Nießner, M., Ferrari, V.: Vid2cad: Cad model alignment using multi-view constraints from videos. arXiv preprint arXiv:2012.04641 (2020) 5
47. Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J.: Structurenet: Hierarchical graph networks for 3d shape generation. arXiv preprint arXiv:1908.00575 (2019) 4
48. Nie, Y., Han, X., Guo, S., Zheng, Y., Chang, J., Zhang, J.J.: Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In: CVPR. pp. 55–64 (2020) 5
49. Nie, Y., Hou, J., Han, X., Niessner, M.: Rfd-net: Point scene understanding by semantic instance reconstruction. In: CVPR. pp. 4608–4618 (June 2021) 2, 5, 6, 7, 9, 10, 11, 12, 13
50. Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S., Zhang, J.J.: Skeleton-bridged point completion: From global inference to local adjustment. arXiv preprint arXiv:2010.07428 (2020) 4
51. Paschalidou, D., Katharopoulos, A., Geiger, A., Fidler, S.: Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In: CVPR. pp. 3204–3215 (2021) 4
52. Pham, Q.H., Nguyen, T., Hua, B.S., Roig, G., Yeung, S.K.: Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: CVPR. pp. 8827–8836 (2019) 4
53. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV. pp. 9277–9286 (2019) 2
54. Ronneberger, O., P.Fischer, Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. LNCS, vol. 9351, pp. 234–241. Springer (2015), `http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a`, (available on arXiv:1505.04597 [cs.CV]) 5
55. Senushkin, D., Belikov, I., Konushin, A.: Decoder modulation for indoor depth completion. arXiv preprint arXiv:2005.08607 (2020) 1
56. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR. pp. 770–779 (2019) 7
57. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: CVPR. pp. 1746–1754 (2017) 2, 5
58. Stutz, D., Geiger, A.: Learning 3d shape completion from laser scan data with weak supervision. In: CVPR. pp. 1955–1964 (2018) 4
59. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: ECCV. pp. 685–702. Springer (2020) 4
60. Tang, J., Chen, X., Wang, J., Zeng, G.: Not all voxels are equal: Semantic scene completion from the point-voxel perspective. arXiv preprint arXiv:2112.12925 (2021) 5

61. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: CVPR. pp. 383–392 (2019) 4
62. Uy, M.A., Kim, V.G., Sung, M., Aigerman, N., Chaudhuri, S., Guibas, L.J.: Joint learning of 3d shape retrieval and deformation. In: CVPR. pp. 11713–11722 (2021) 5
63. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. TOG **36**(4), 1–11 (2017) 1
64. Wang, W., Yu, R., Huang, Q., Neumann, U.: Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In: CVPR. pp. 2569–2578 (2018) 4
65. Wang, X., Ang Jr, M.H., Lee, G.H.: Cascaded refinement network for point cloud completion. In: CVPR. pp. 790–799 (2020) 4
66. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NeurIPS. pp. 82–90 (2016) 7
67. Wu, R., Zhuang, Y., Xu, K., Zhang, H., Chen, B.: Pq-net: A generative part seq2seq network for 3d shapes. In: CVPR. pp. 829–838 (2020) 4
68. Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J.: Mlcvnet: Multi-level context votenet for 3d object detection. In: CVPR. pp. 10447–10456 (2020) 2
69. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018) 4
70. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3d instance segmentation on point clouds. arXiv preprint arXiv:1906.01140 (2019) 4
71. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 3DV. pp. 728–737. IEEE (2018) 4
72. Zhang, C., Cui, Z., Zhang, Y., Zeng, B., Pollefeys, M., Liu, S.: Holistic 3d scene understanding from a single image with implicit representation. In: CVPR. pp. 8833–8842 (2021) 5
73. Zhong, M., Chen, X., Chen, X., Zeng, G., Wang, Y.: Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In: ICME (2022) 4