JIMR: Joint Semantic and Geometry Learning for Point Scene Instance Mesh Reconstruction

Qiao Yu, Xianzhi Li✉, Yuan Tang, Jinfeng Xu, Long Hu, Yixue Hao, and Min Chen, *Fellow, IEEE*

Abstract—Point scene instance mesh reconstruction is a challenging task since it requires both scene-level instance segmentation and instance-level mesh reconstruction from partial observations simultaneously. Previous works either adopt a detection backbone or a segmentation one, and then directly employ a mesh reconstruction network to produce complete meshes from incomplete instance point clouds. To further boost the mesh reconstruction quality with both local details and global smoothness, in this work, we propose JIMR, a joint framework with two cascaded stages for semantic and geometry understanding. In the first stage, we propose to perform both instance segmentation and object detection simultaneously. By making both tasks promote each other, this design facilitates subsequent mesh reconstruction by providing more precisely-segmented instance points and better alignment benefiting from predicted complete bounding boxes. In the second stage, we propose a complete-then-reconstruct procedure, where the completion module explicitly disentangles completion from reconstruction, and enables the usage of pre-trained weights of existing powerful completion and reconstruction networks. Moreover, we propose a comprehensive confidence score to filter proposals considering the quality of instance segmentation, bounding box detection, semantic classification, and mesh reconstruction at the same time. Experiments show that our proposed JIMR outperforms state-of-the-art methods regarding instance reconstruction qualitatively and quantitatively.

Index Terms—Instance Mesh Reconstruction, 3D Scene Understanding, 3D Reconstruction

1 Introduction

Instance mesh reconstruction (IMR) from a 3D point scene is a crucial step towards holistic 3D scene understanding for enriching various real-world applications, such as robot navigation, games, AR/VR, and interior design, etc. This task aims to not only understand the semantic information of each object but also recover their geometries from partial observations. In short, IMR is a multi-objective task that unifies object recognition from a 3D scene and mesh reconstruction from a partially observed point cloud together.

Most previous scene understanding methods only focus on one or two tasks, e.g., semantic/instance segmentation [1], [2], [3], [4], [5], [6], [7], semantic scene/instance completion [8], [9], [10] or scene-level surface reconstruction [11], [12], [13], [14], thus lacking abilities to provide both comprehensive scene understanding and instance-level high-resolution mesh reconstruction from partial observations.

While inevitably involving the above single-task methods, existing IMR works especially focus on how to integrate these backbones more effectively and efficiently, so as to explore both semantic and geometry information and further unify object recognition and mesh reconstruction together. In other words, existing IMR works aim to propose a general and refined framework, where sub-modules (backbones) can be flexibly changed. For instance, the pioneering work RfD-Net [15] first proposes a detect-then-segment process and reconstructs complete meshes directly from incomplete point clouds. However, as shown in Figure 1, RfD-Net tends to generate abundant and thick structures. To reduce false positive instance proposals and handle the ambiguity of incomplete scans, DIMR [16] conducts instance segmentation and then implicitly completes missing geometries by regressing latent codes of complete shapes from incomplete points, then followed by mesh reconstruction. See again Figure 1, though the reconstructed mesh by DIMR has no obvious abundant structures, the global smoothness and completeness are worse.

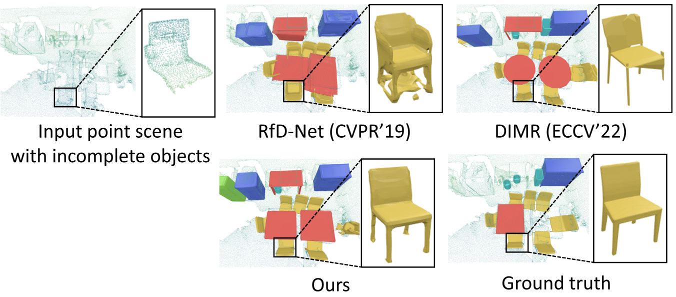

Fig. 1. Comparing the instance mesh quality of RfD-Net [15], DIMR [16] and our proposed JIMR given input point scene with incomplete objects. Unlike RfD-Net with abundant and thick structures or DIMR with a low-poly abstract style, our method yields the reconstructed mesh with local details and smooth surfaces.

In general, the reasons why existing works are not so good can be mainly categorized into two aspects. The first one is the *error accumulation in cascaded tasks*. That is, the accuracy and completeness of segmented points directly affect the quality of the subsequent mesh reconstruction. Intuitively, a poorly segmented instance suffering from false positive points from nearby objects tends to yield a noisy

- *Q. Yu, X. Li, Y. Tang, J. Xu, L. Hu, Y. Hao and M. Chen are with Huazhong University of Science and Technology, E-mail: {qiaoyu_epic, xzli, yuantang,jinfengxu,longhu,yixuehao, minchen2012}@hust.edu.cn.*

Corresponding author: Xianzhi Li

reconstructed mesh, while false negative points would cause missing geometry, thus increasing the difficulty of shape completion. The second one is the *misalignment between the incomplete point clouds and the complete ground-truth meshes*, thus obstructing training an effective shape completion network; see Figure 2 for an illustration. Specifically, a segmented instance needs to be canonically transformed from a scene-level global coordinate system to its object-level local one for training a shape completion (or reconstruction) network. To realize such a transformation, an accurate complete bounding box (abbr. bbox) predicted from a partial instance is required. However, the structure ambiguity caused by incomplete point observations leads to inaccurate bbox, thus resulting in the misalignment with the ground-truth shape. Note that, we experimentally find that feeding ground-truth segmentation and detection results to the completion and reconstruction modules significantly improves reconstruction quality, which proves the rationality of the above analysis.

To overcome these shortcomings, we are motivated to develop our **JIMR**, a new approach by exploiting instance segmentation, object detection, shape completion and mesh reconstruction **J**ointly for point scene **I**nstance **M**esh **R**econstruction. We design our JIMR with two cascaded stages for semantic and geometry understanding, respectively. In the first stage, to ensure an accurate instance segmentation for subsequent mesh reconstruction, we propose to perform both instance segmentation and object detection by formulating two parallel branches to obtain point-wise results. In this way, we can not only get both (partial) instance points and (complete) bboxes in one stage, but also make these two highly-related tasks promote each other. Next, we group and merge these point-wise results into instance-wise ones, which are then transformed into their local canonical coordinate systems benefiting from the predicted complete bboxes.

In the second stage, instead of reconstructing meshes directly from partial segmented points, we propose to disentangle this complex task into two sub-tasks: (i) to generate complete instance points from partial observations first, and then (ii) to reconstruct meshes from predicted complete points. To do so, we introduce a simple yet effective point cloud completion network [17] followed by a mesh generation network [18] to produce the final reconstructed complete instance meshes. In addition, in this stage, we propose to regress multiple kinds of confidence scores, including instance segmentation scores, bbox prediction scores, mesh reconstruction scores, as well as semantic classification scores, so as to filter out low-confidence samples. Overall, we list our technical contributions as follows:

(i) We formulate JIMR with a joint instance segmentation and object detection backbone, which makes both tasks promote each other to achieve precise instance segmentation and complete bbox prediction within a single stage, and the predicted complete bboxes facilitate better-aligned inputs for the subsequent completion and reconstruction modules.

(ii) We introduce a complete-then-reconstruct strategy where the completion module explicitly disentangles completion from reconstruction, and enables the usage of pre-trained weights of existing powerful completion and reconstruction networks to lower the training difficulty without latent space regularization and produce good results with short training time.

(iii) We propose a series of effective confidence scores for low-confidence sample filtering considering the quality of instance segmentation, bounding box detection, semantic classification and mesh reconstruction at the same time.

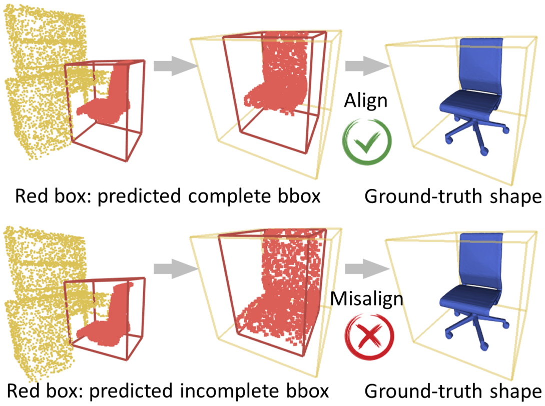
Align
Red box: predicted complete bbox
Ground-truth shape
Misalign
Red box: predicted incomplete bbox
Ground-truth shape

Fig. 2. A predicted complete bounding box covering both the observed partial points and the missing parts (top) from partial scans provides better alignment with ground truths via canonical transformation, while an incomplete bounding box only covering the partial observations (bottom) leads to misalignment, thus affecting the reconstruction performance.

We demonstrate the effectiveness and superiority of our JIMR compared to SOTA methods both quantitatively and qualitatively with a significant margin on public benchmark datasets, and verify the contributions of all the above core designs by extensive ablation experiments.

2 Related Work

Instance Mesh Reconstruction. 3D reconstruction from large scenes [11], [12], [13], [14] often yields incomplete instances caused by view constraints or occlusions between nearby objects. To recover occluded instances and reconstruct the associated meshes, instance mesh reconstruction given point scenes has been explored in recent years. RfD-Net [15] was the pioneer to propose a detect-segment-reconstruct process by first employing a detection network [19] to obtain bounding box proposals, and then using a small segmentation network to filter out background points, followed by an Occ-Net [18] to produce meshes. Recently, DIMR [16] replaced the detection backbone with a segmentation one to reduce false positives. Instead of directly regressing implicit fields as in RfD-Net, DIMR regressed the latent codes of complete instances and then used a pre-trained shape decoder [20] to output meshes.

Note that, both RfD-Net and DIMR need to use the detected incomplete instances (points) to train a reconstruction network to produce the associated complete meshes. Hence, the input incomplete instances should be aligned with the ground-truth meshes for better feature learning. To do so, both methods conducted a canonical transformation for instance points based on predicted bounding boxes. However, DIMR's canonical transformation is performed directly using incomplete instance bbox, i.e., the bbox that only covers the observed partial points; see the red 3D box in the bottom part of Figure 2. Yet, the ground truths are complete, thus leading to misalignment. In our work, we propose to conduct the transformation based on the predicted complete instance bbox (i.e. the bbox that covers both the observed partial points and the missing parts; see the top row in Figure 2), so that the transformed instance points will be better aligned with the ground truth meshes, helping the network to learn better.

3D Instance Segmentation and Object Detection. Instance segmentation and object detection are two important and relevant tasks in 3D scene understanding. Existing works either choose one task as the focus and the other as an auxiliary, or treat both tasks equally. In general, detection-for-segmentation methods [15], [21], [22] predict bounding boxes first, and then in each box, they segment foreground points as the instance segmentation results. While segmentation-for-detection methods [23], [24], [25] conduct foreground or semantic segmentation first, and then regress a bounding box proposal for each detected foreground point. These proposals are filtered by non maximum suppression (NMS), and then may be fed into further steps for a second-stage refinement. Joint segmentation-and-detection methods [26], [27], [28], [29] regard the two tasks equally and leverage the advantages of both. In these approaches, two parallel branches for segmentation and detection are designed. For each segmented instance, they calculate its bounding box by averaging boxes of all or top-K points of it, instead of using NMS. This kind of methods have shown that sharing features for segmentation and detection branches promotes both tasks and are suitable for tasks that require both segmentation and detection results, like our instance mesh reconstruction.

Single Object Completion. Single object completion aims to predict complete single objects from incomplete inputs. Voxel-based methods [30], [31] usually use 3D CNNs, which are memory-consuming due to their cubic complexity. Point-cloud-based methods [17], [32], [33], [34], [35], [36], [37], [38], [39] are usually based on modified PointNet++ [40] or transformers [41]. While most methods follow a supervised learning style that takes pairs of complete and partial shapes for training, some recent works [42], [43] adopt auto-regressive models to achieve impressive performance, but they are expensive regarding time and memory. In this work, we propose to use a simple point cloud completion network (i.e., PCN [17]) followed by an implicit function-based mesh reconstruction network (i.e., Occ-Net [18]), to balance performance, time, and memory consumption.

3 Method

3.1 Overview

We illustrate the network architecture of our JIMR in Figure 3. Generally, our pipeline consists of two cascaded stages: *point-wise joint segmentation & detection* and *instance-wise completion & reconstruction*, with an *instance merging and transforming* module for connecting the two stages. Note that the merging and transforming module involves no network training. Given a 3D scene with N points as input, the first stage employs a sparse 3D U-Net [16] for feature extraction followed by joint instance segmentation and 3D bounding box generation. For the instance segmentation branch, it outputs per-point semantic logits $\mathbf{l}$ and per-point offsets $\mathbf{o}$ pointing to instance centers. For the bounding box generation branch, it outputs per-point complete 3D box parameters $\mathbf{b}$, which are then decoded into the standard 7-dim box representation $\mathbf{b}'$. Based on $\mathbf{o}$ and $\mathbf{l}$, we first group N points into L proposal instances, then merge (i.e., average) all the box parameters $\mathbf{b}'$ of points belonging to each grouped proposal to obtain instance boxes $\mathbf{B}$. Note that, the predicted boxes $\mathbf{B}$ are complete even if predicted from partial inputs, thus assisting in better alignment. Next, randomly sampled instance proposal points are transformed into their local canonical coordinate systems using $\mathbf{B}$, and then sent to the second stage for completion and reconstruction.

The second stage starts with an instance encoder [17] to extract proposal-level features $\mathbf{F}$, which are then used in two parallel branches. In the confidence score branch, four MLPs are used to predict segmentation, semantic, bbox and mesh scores, which are multiplied to form the final confidence scores for proposal filtering. In the completion and reconstruction branch, we first use a point cloud completion decoder [17] to create completed proposal point clouds $\mathbf{I}^C$, and then feed them to a mesh generation network [18] to predict occupancy values. The final instance meshes are reconstructed via the Marching Cubes algorithm [44].

3.2 Point-wise Joint Segmentation and Detection

To obtain object-level instance proposals for subsequent mesh reconstruction, existing works adopt either a 3D bbox detection backbone or an instance segmentation one. For example, RfD-Net [15] puts more focus on bbox detection and only conducts light-weight foreground point segmentation for each proposed box. However, the segmentation quality is relatively low due to the small size of the segmentation network and the error introduced from previous detection. DIMR [16], on the other hand, takes a segmentation backbone in the first stage, canonically transforms instance points, and predicts residual bboxes and shape latent codes in parallel in the second stage. However, the canonical transformation happens before predicting complete bboxes, which means that the shape latent codes are regressed with misaligned points.

In our work, to realize both accurate instance segmentation and complete bounding box regression, we propose joint object segmentation and detection motivated by [26], [27], [28], [29]. Specifically, we follow [16] to first voxelize the input point scene $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ with N points and adopt a sparse 3D U-Net as the backbone for feature extraction. The voxel-wise output features are mapped back to point-wise features, which are then fed into two parallel MLP branches: instance segmentation and 3D bbox generation.

Instance segmentation branch. We follow the common routine [4], [6], [16], [26] to design this branch with two output heads: an offset head and a semantic head. The offset head predicts the offset $\mathbf{o}_i = (\Delta x_i, \Delta y_i, \Delta z_i)$ of each point $\mathbf{p}_i$, denoting the vector from $\mathbf{p}_i$ to its instance center. The semantic head predicts the classification logits $\mathbf{l}_i \in \mathbb{R}^C$ for $\mathbf{p}_i$, where C is the number of semantic categories. The losses used for the offset head consist of a norm loss $L_{\text{norm}}^{\text{offset}}$ and a direction loss $L_{\text{dir}}^{\text{offset}}$, where the former is the $L1$ distance

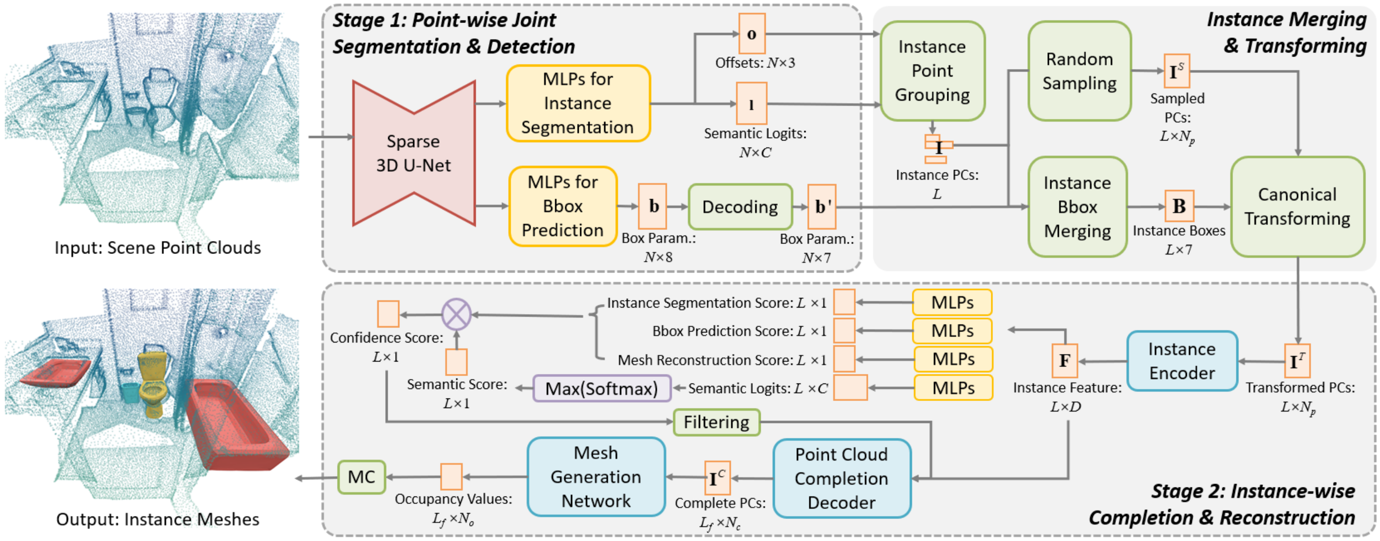

Fig. 3. Overview of our approach. In Stage 1, the network predicts point-wise offsets $\mathbf{o}$, semantic logits $\mathbf{l}$, and bounding box parameters $\mathbf{b}$. A grouping operation clusters N points into L instances $\mathbf{I}$. Decoded point-wise box parameters $\mathbf{b}'$ are merged by instance to form the instance boxes $\mathbf{B}$, with which the randomly sampled N_p points of each instance are canonically transformed into local coordinate systems as $\mathbf{I}^T$. In Stage 2, an encoder extracts instance features $\mathbf{F}$, followed by a point cloud completion decoder and a mesh generation network to reconstruct meshes. A confidence score branch is used to learn various scores for filtering out low-quality instance proposals.

between the predicted and ground-truth offsets, and the latter is the opposite of the product of predicted and ground-truth offset direction vectors. For more details, please refer to [4]. The loss used for the semantic head is the cross entropy loss referred as $L_{\text{cls}}^{\text{sem}}$.

3D bbox generation branch. In this branch, we regress bbox parameters for each foreground point $\mathbf{p}_i$ (i.e. points that belong to any ground-truth instance). Specifically, we follow FCAF3D [45] to predict the distances from $\mathbf{p}_i$ to its corresponding instance bbox's six surfaces $(d_1, d_2, d_3, d_4, d_5, d_6)$ instead of directly predicting box centers (x, y, z) and scales (w, l, h). Note that, the angle representation of FCAF3D's Mobius box parametrization is not suitable for our case. It is originally proposed to deal with ambiguous orientations of symmetric shapes, and can only produce an angle $r \in (-\pi/2, \pi/2)$ due to its inverse tangent operation. However, we need $r \in [-\pi, \pi)$. Therefore, we use bin-based angle representation [16], [23] instead, by equally dividing the range $[-\pi, \pi)$ into 12 bins and predicting which bin r lies in, and then predicting a residual angle offset inside the bin. In total, we predict the following eight box parameters: $(d_1, d_2, d_3, d_4, d_5, d_6, r_{\text{bin}}, r_{\text{res}})$. The eight parameters are then decoded to the standard 7-dim box representation as (x, y, z, w, l, h, r) for loss calculation. Please refer to our supplementary file for the detailed decoding process. We use three losses for the bbox generation branch: rotated bounding box IoU loss $L_{\text{iou}}^{\text{bbox}}$, angle bin classification loss $L_{\text{cls}}^{\text{angle}}$ (cross entropy loss), and residual angle regression loss $L_{\text{reg}}^{\text{angle}}$ (smooth $L1$ loss).

The overall loss function of the first stage is the weighted sum of the above-mentioned losses:

$$\begin{aligned} L_{\text{stage1}} &= w_1 \times L_{\text{norm}}^{\text{offset}} + w_2 \times L_{\text{dir}}^{\text{offset}} + w_3 \times L_{\text{cls}}^{\text{sem}} \\ &+ w_4 \times L_{\text{iou}}^{\text{bbox}} + w_5 \times L_{\text{cls}}^{\text{angle}} + w_6 \times L_{\text{reg}}^{\text{angle}}, \end{aligned} \tag{1}$$

where the weights $\{w_i\}_{i=1}^{6}$ are used to balance each term.

3.3 Instance Merging and Transforming

The first stage outputs point-wise features in the global coordinate system, while the second stage needs instance-wise data in the local canonical coordinate systems. Therefore, as shown in Figure 3, we perform instance merging and transforming on the point-wise outputs of Stage 1 to prepare instance-wise inputs for Stage 2. In detail, given point-wise offsets $\mathbf{o}$ and semantic labels $\max(\text{softmax}(\mathbf{l}))$, we first group the N points into L proposal instances as $\mathbf{I} = \{I_i\}_{i=1}^{L}$. Then, to get the bbox for each proposal I_i, we average the 7-dim box parameters of all points in I_i to get the merged mean box $B_i = \{x_i, y_i, z_i, w_i, l_i, h_i, r_i\}$. Unlike detection-based methods that first propose numerous bboxes and then filter them by NMS, our "Group and Merge" operation naturally keeps only one bbox for each segmented proposal, where averaging bbox parameters may also improve robustness to yield more accurate bboxes. Since each instance I_i contains different numbers of points, but the networks in Stage 2 require fix-number point clouds as input, we thus randomly sample N_p points from each instance as $\mathbf{I}^S = \{I_i^S\}_{i=1}^{L}$. We adopt sampling with replacement in case some instances contain points fewer than N_p. With B_i, we transform I_i^S into its local canonical coordinate system by moving it to B_i's center $\mathbf{c}_i = \{x_i, y_i, z_i\}$, rotating it around z-axis for $-r_i$, and then dividing it by its maximum scale $\max(\mathbf{s}_i)$ where $\mathbf{s}_i = \{w_i, l_i, h_i\}$ so that coordinates of all points are within $[-0.5, 0.5]$. The canonically transformed proposal point instances are referred as $\mathbf{I}^T = \{I_i^T\}_{i=1}^{L}$.

3.4 Instance-wise Completion and Reconstruction

Existing works [15], [16] directly use mesh generation networks [18], [20] to create instance meshes from partial point clouds. However, the employed mesh generation networks are originally designed for processing complete point clouds without the capability of shape completion. Though DIMR has proposed to disentangle completion and mesh

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3398737

generation, it only performs implicit completion in latent space. Motivated by existing powerful shape completion methods, we propose to explicitly incorporate a small completion network before mesh reconstruction. In this way, we disentangle the challenging mesh reconstruction task into two sub-tasks: (i) to generate complete instance points from partial observations first, and then (ii) to reconstruct meshes from predicted complete points. Compared with reconstructing meshes from partial observations in one step, this disentanglement scheme effectively relieves the burden of network learning.

To balance performance, time, and memory consumption, we use a simplified PCN [17] for explicit shape completion, followed by an Occ-Net [18] for mesh reconstruction, both of which are simple yet effective. Particularly, experiments found that by using the pre-trained weights of PCN and Occ-Net, the network can produce reasonable meshes with very few epochs of the second-stage training. Specifically, as shown in Figure 3, the L canonically transformed instance point clouds $\mathbf{I}^T$ are fed into an instance encoder. We directly use the encoder in PCN [17], which is a modified PointNet [46] and outputs instance features $\mathbf{F} \in \mathbb{R}^{L \times D}$, where D is the feature dimension. After that, $\mathbf{F}$ is fed into two branches: one for mesh generation and the other for confidence score regression.

Mesh generation branch. In this branch, we first feed $\mathbf{F}$ into a decoder for point cloud completion. The original PCN decoder contains a coarse MLP-based decoder to output 1024 points per object and a detailed folding-based decoder for expanding the point number by 16 times. We only adopt the coarse one since 1024 points are already enough for Occ-Net to yield satisfying reconstruction results. After that, the predicted complete instance point clouds $\mathbf{I}^C = \{I_i^C\}_{i=1}^L$ are fed into an Occ-Net to predict implicit fields for mesh generation. The Occ-Net adopts an encoder-decoder structure. The encoder outputs a latent code $\mathbf{z}$ for each object. The decoder outputs the occupancy values of points sampled inside a canonical cube (a cube located at the origin and with a length of 1.0), which indicates whether they are inside or outside of the shape surface. To guide the learning of Occ-Net, besides the commonly-used binary cross entropy (BCE) loss to supervise the occupancy values, inspired by [16], [47], we further encourage the latent code $\mathbf{z}$ to be similar to a pre-trained teacher Occ-Net, which is trained using points sampled from ground-truth meshes.

Hence, the losses used in this branch consist of three parts: the completion loss $L_{\text{CD}}^{\text{comp}}$, the Occ-Net latent code loss $L_{\text{latent}}^{\text{mesh}}$, and the occupancy value loss $L_{\text{cls}}^{\text{mesh}}$ for mesh prediction. $L_{\text{CD}}^{\text{comp}}$ is the Chamfer distance of predicted and ground-truth complete point clouds. $L_{\text{latent}}^{\text{mesh}}$ is the the smooth L1 loss of the predicted and teacher latent code $\mathbf{z}$. And $L_{\text{cls}}^{\text{mesh}}$ is the BCE loss of occupancy values.

Confidence score branch. The instance proposals produced by the first stage contains too many false positive samples. A commonly-used routine during inference is to filter them by NMS and a pre-defined score threshold, both based on confidence scores. To achieve this, we predict the confidence scores for each proposal via a MLP-based branch. Unlike DIMR [16] that only uses a segmentation score, we argue that the semantic classification, bounding box prediction and mesh reconstruction quality also influence the results. Hence, as shown in Figure 3, we design four MLP-based heads to regress these scores. The final confidence score is the product by multiplying the four scores. Compared with other operations such as addition or power on the four scores, the multiply operation acts as a one-vote-veto mechanism that requires all scores to be higher than a threshold, thus requiring each task to be learned well and naturally preventing a few extremely high scores from dominating the final score.

To guide the learning of these heads, we use BCE loss to supervise the segmentation score by the IoU between segmented and groud-truth instance points, bbox score by the IoU between predicted and ground-truth bounding boxes, and the mesh score by the accuracy of predicted occupancy values and ground-truth ones. For the semantic classification head, we supervise it by cross entropy loss and take the probability of the final class calculated by softmax as the semantic score. The overall loss function of the second stage is the weighted sum of the above-mentioned losses:

$$\begin{aligned} L_{\text{stage2}} = w_7 \times L_{\text{CD}}^{\text{comp}} + w_8 \times L_{\text{latent}}^{\text{mesh}} + w_9 \times L_{\text{cls}}^{\text{mesh}} \\ + w_{10} \times L^{\text{seg\_score}} + w_{11} \times L^{\text{cls\_score}} \\ + w_{12} \times L^{\text{box\_score}} + w_{13} \times L^{\text{mesh\_score}}. \end{aligned} \tag{2}$$

3.5 Network Training and Inference

We follow the same training and inference pipelines as DIMR and RfD-Net. Specifically, since Stage 1 is the foundation of Stage 2, we first train Stage 1 till converging, and then jointly train both stages based on pre-trained PCN and Occ-Net weights. During the second-stage training, we only conduct point cloud completion and occupancy value prediction for instances with ground-truth segmentation IoU bigger than 0.5. Note that we freeze the Occ-Net decoder during training as this works better experimentally. Please refer to our supplementary file for more details. During inference, we use NMS, a minimum-point-number threshold of 100, and a confidence score threshold of 0.01 to filter proposals. Note that this confidence score threshold is much lower than that of DIMR, since we multiply four scores as the final one. The selected proposals are sent to the completion and reconstruction branch to predict instance occupancy grids, from which meshes are generated by Marching Cubes [44]. We set the resolution of occupancy grids to be 32 following RfD-Net. *We shall release the code and trained networks upon publication of this work.*

4 Experiments and Results

4.1 Experimental Settings

Datasets. We follow DIMR [16] to evaluate our method using three datasets: ScanNet v2 [48], ShapeNet [49], and Scan2CAD [50]. ScanNet v2 consists of 1513 real-world indoor scene scans, from which we use the point clouds as inputs, and the point-wise semantic and instance labels for our first-stage segmentation supervision. Scan2CAD aligns ScanNet instances with ShapeNet model meshes and provides corresponding semantic labels and 3D bounding boxes for our first-stage detection supervision. For the second-stage supervision, we randomly sample points from

TABLE 1
Comparisons on mesh reconstruction quality. We report mAP measured with IoU, CD and LFD at different thresholds. A higher value indicates a better result and the best results are marked in bold. Clearly, our JIMR outperforms others on most metrics significantly.

	IoU@0.25	IoU@0.5	CD@0.1	CD@0.047	LFD@5000	LFD@2500
RfD-Net (CVPR'21) [15]	42.52	14.35	46.37	19.09	28.59	7.8
DIMR (ECCV'22) [16]	46.34	12.54	**52.39**	25.71	29.47	8.55
JIMR (Ours)	**49.20**	**15.68**	50.67	**25.99**	**31.33**	**11.50**

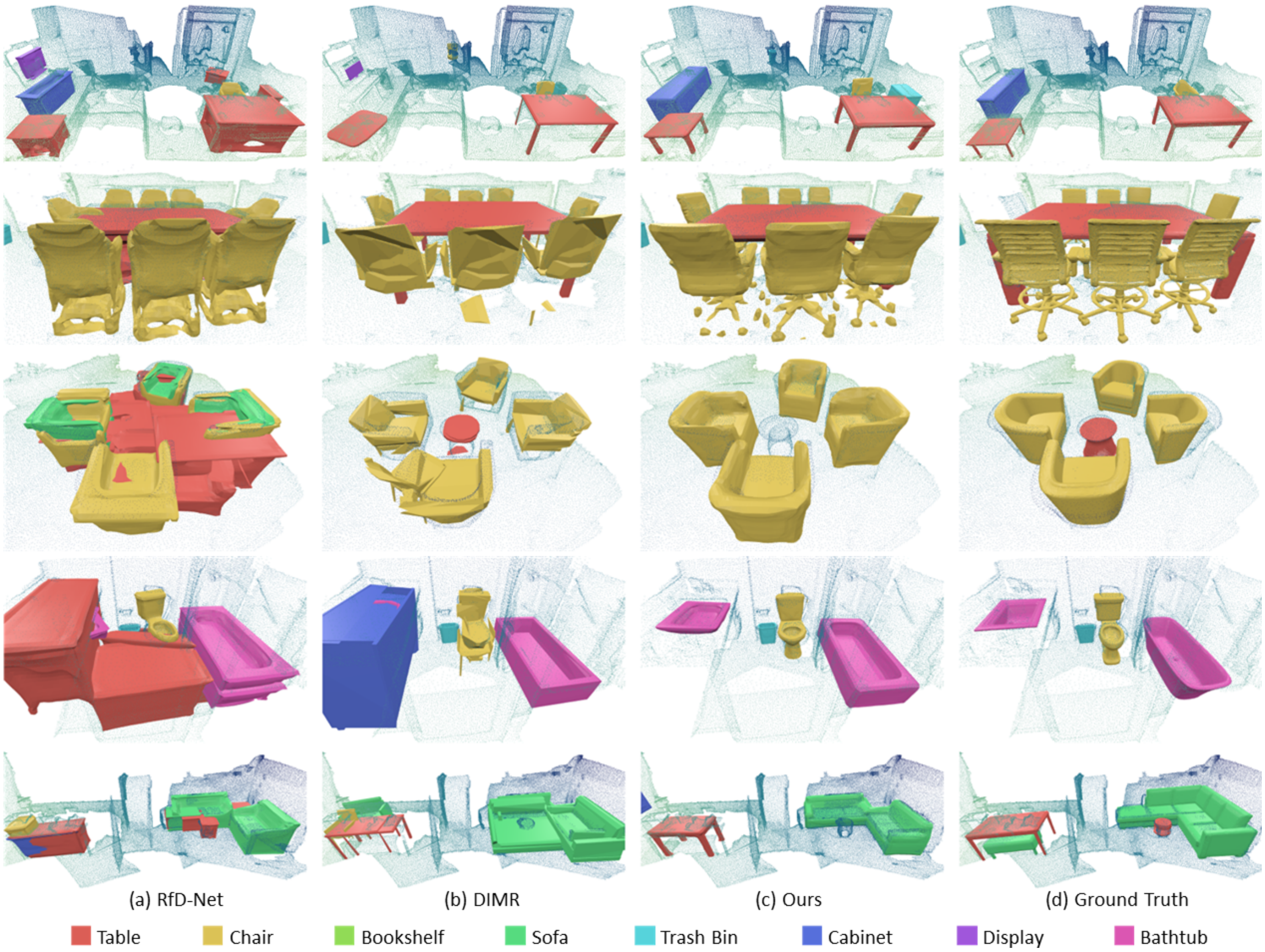

Fig. 4. Qualitative results of instance mesh reconstruction on ScanNet v2. Note that the results of RfD-Net are obtained by running their official code and the results of DIMR are provided by their authors.

the ShapeNet models as ground-truth complete point clouds to train our simplified PCN, and follow [15], [18] to create ground-truth occupancy values to train our Occ-Net. We follow the compatible label system of DIMR [16] to handle the inconsistency between ScanNet semantic labels, ScanNet instance labels, and Scan2CAD mesh labels. We use the same train-test data split following RfD-Net and DIMR.

Metrics. We follow [16] to use 3D intersection over Union (IoU), Chamfer Distance (CD) and Light Field Distance (LFD) as evaluation metrics. 3D IoU reflects the similarity of shape occupancy grids, CD measures the distances between mesh surface points, and LFD focuses more on visual appearance. For a thorough comparison, we adopt all the three metrics with different thresholds to determine how well the predicted meshes match their associated ground truths, and report the mean average precision (mAP) over all classes. Please refer to [16] for more details.

Implementation details. We set most hyper parameters following DIMR [16], including voxel size, learning rates, etc. More details are provided in the supplementary file. We train the first stage for 300 epochs, and then jointly train both stages for only 4 epochs. The training takes around 60 hours for the first stage and 2 hours for the second.

4.2 Comparisons to State-of-the-Arts

Quantitative comparisons. We compare our work on instance mesh reconstruction with state-of-the-art works RfD-Net [15] and DIMR [16] using the same test set with 311 scenes. Table 1 summarizes the quantitative comparisons,

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3398737

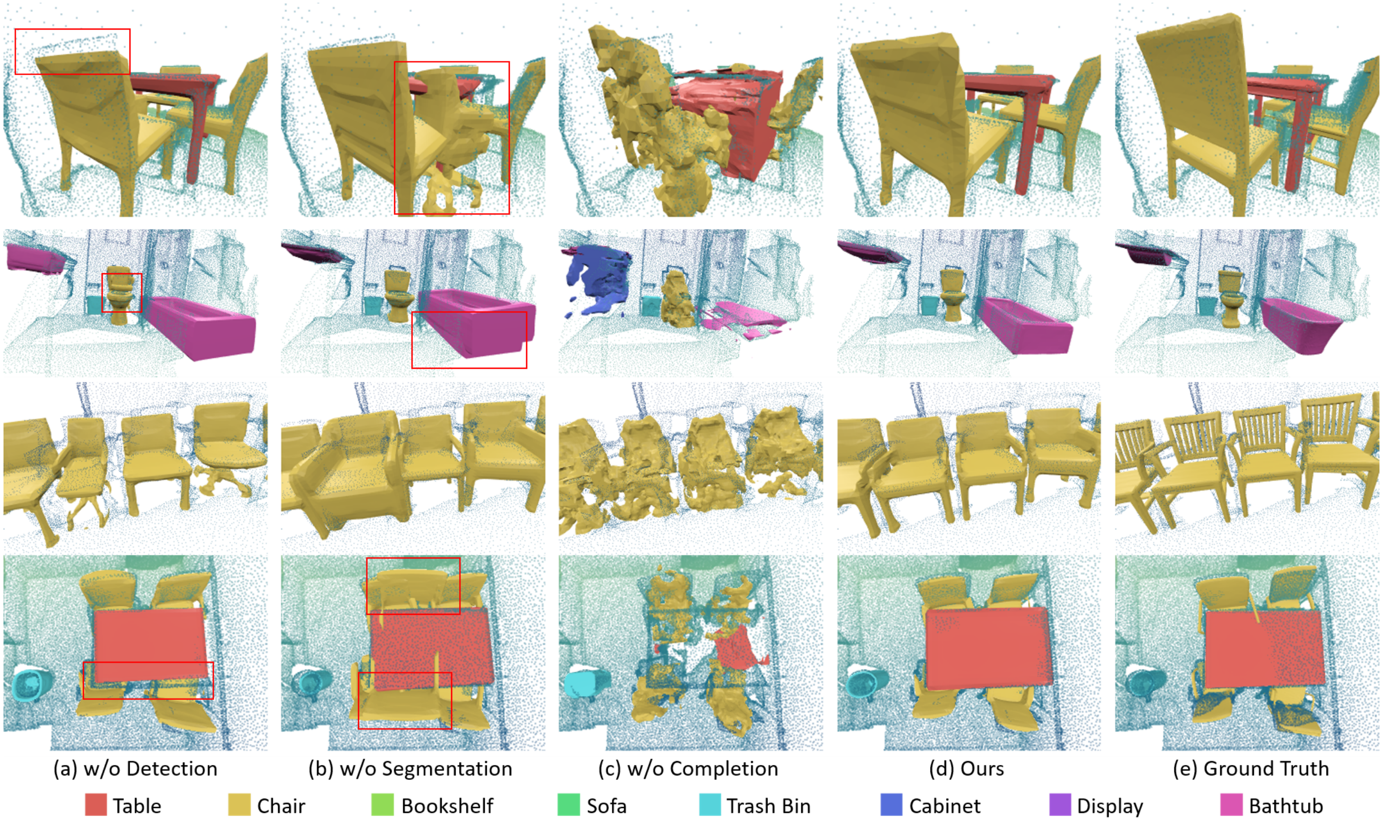

Fig. 5. Ablation comparisons of our complete joint learning pipeline against removing detection, segmentation or completion module.

where we report mAP measured with IoU, CD and LFD at different thresholds. Note that, a higher value indicates a better result. For IoU, a higher threshold is more strict, while for CD and LFD, a lower threshold is more strict. The results of RfD-Net and DIMR are taken from the paper of DIMR. From the results, we can see that our JIMR achieves the highest values on five out of six metrics. Particularly, for strict metrics like IoU@0.5, CD@0.047 and LFD@2500, our produced mAP scores are all higher than others.

To be honest, our method has no obvious advantage in terms of CD compared to DIMR. We think that this is caused by the reconstruction network. In DIMR, BSP-Net [20] is used for mesh reconstruction. It has a natural advantage regarding CD since it produces thinner and more compact meshes with delicate local structures so that the mesh vertices can be closer to ground truths, resulting in better CD scores. Nevertheless, the Occ-Net [18] employed by us is superior in IoU, because it produces relatively thicker meshes. It's more likely to occupy target voxels to obtain a higher IoU, but a lower CD. As for LFD that mainly focuses on visual effects, our JIMR reaches the highest scores. This is consistent with the visual comparisons in Figure 4 that our method produces meshes with higher quality.

Qualitative comparisons. Figure 4 illustrates scene-level qualitative results. Unlike RfD-Net with thick and redundancy structures (a) or DIMR with a low-poly abstract style (b), our method produces more realistic object meshes with better local details (c), especially for toilets and office chairs. More results are provided in the supplemental file.

TABLE 2
Ablation study of proposed modules. The first row is our full pipeline, and for other rows, we only remove one module (marked by ✗) at a time from our full pipeline.

Detection	Segmentation	Completion	Confidence scores				CD@0.1	CD@0.047
			Seg	Cls	Bbox	Mesh		
-	-	-	-	-	-	-	**50.67**	**25.99**
✗		-	-	-	-	-	48.70	25.77
-	✗	-	-	-	-	-	49.44	24.53
-	-	✗	-	-	-	-	22.97	7.04
-	-	-	✗	-	-	-	48.95	25.23
-	-	-	-	✗	-	-	47.41	22.58
-	-	-	-	-	✗	-	49.05	25.38
-	-	-	-	-	-	✗	48.98	25.23

4.3 Ablation Study

We conduct ablation studies on the full test set to verify the contribution of our major components by removing one component from our full pipeline at a time to observe the results. We present the quantitative results in Table 2, where the first row is our full pipeline with all components. We also provide some qualitative comparisons in Figure 5.

Effect of joint segmentation and detection: When removing 'Detection', it means that we follow DIMR to adopt a segmentation backbone instead of a joint one, and only use the calculated incomplete bounding box based on the segmented partial instance points for shape alignment. It causes a performance drop as shown in Table 2, and tends to yield worse reconstruction quality regarding object sizes or shape details, as shown in Figure 5 (a).

When removing 'Segmentation', it means that we fol-

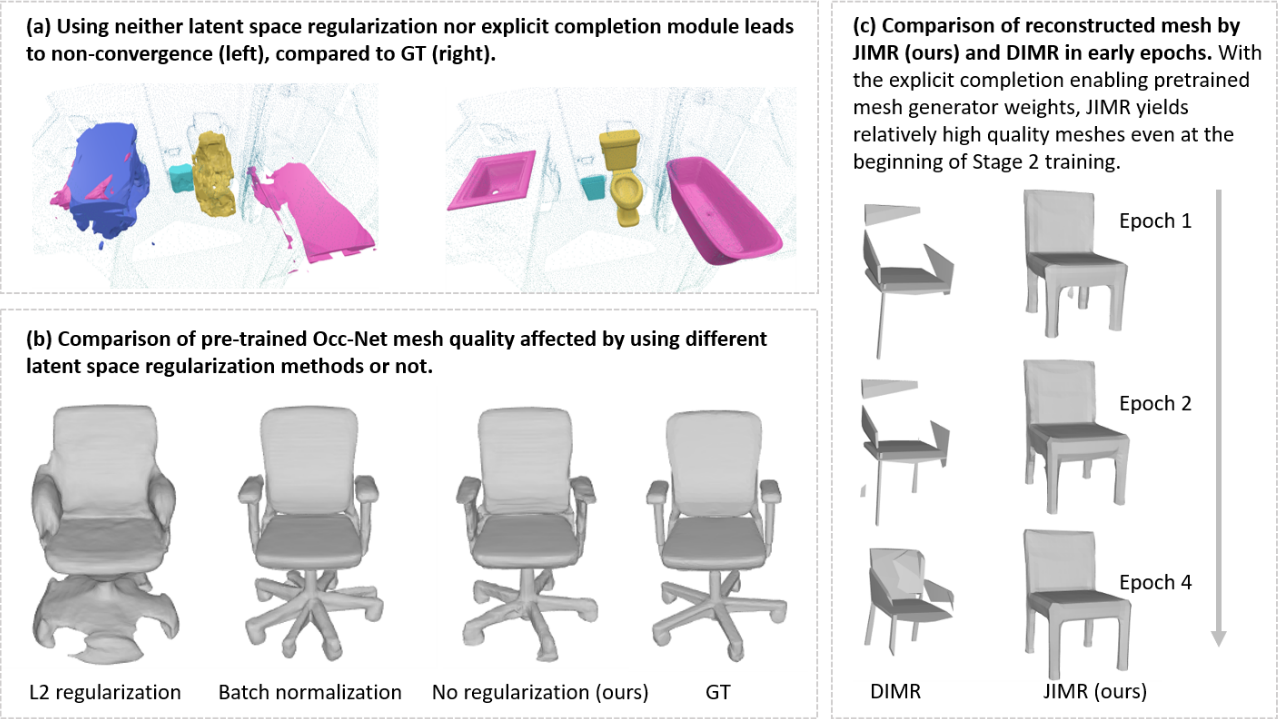
(a) Using neither latent space regularization nor explicit completion module leads to non-convergence (left), compared to GT (right).
(b) Comparison of pre-trained Occ-Net mesh quality affected by using different latent space regularization methods or not.
L2 regularization
Batch normalization
No regularization (ours)
GT
(c) Comparison of reconstructed mesh by JIMR (ours) and DIMR in early epochs. With the explicit completion enabling pretrained mesh generator weights, JIMR yields relatively high quality meshes even at the beginning of Stage 2 training.
Epoch 1
Epoch 2
Epoch 4
DIMR
JIMR (ours)

Fig. 6. (a) When using neither latent space regularization nor explicit completion module, the network cannot converge. (b) When applying different latent space regularization methods, the mesh quality is still worse than ours with explicit completion but without any constraint. (c) The explicit completion module enables pre-trained mesh generator weights, and thus making the training process faster and easier.

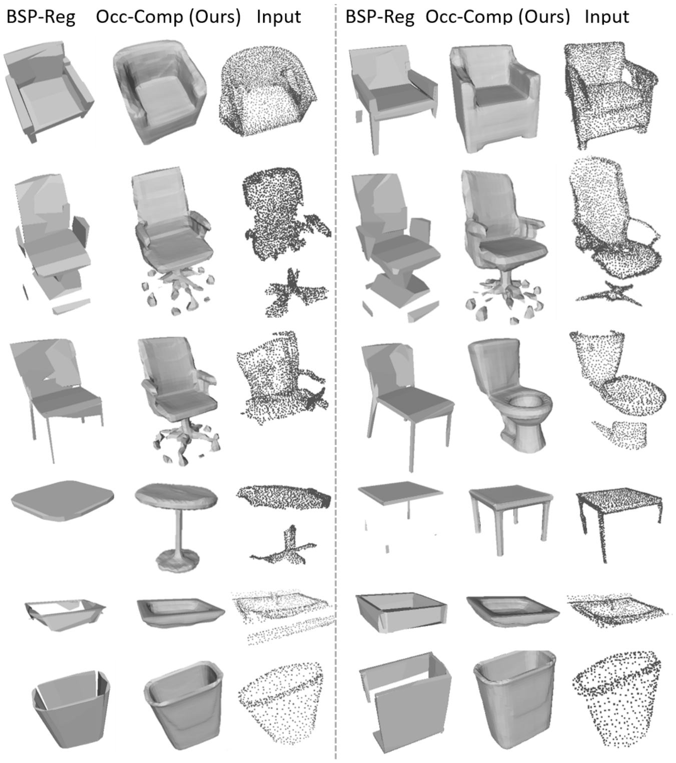

Fig. 7. Qualitative ablation comparing 'BSP-Net + latent regularization (BSP-Reg)' with our 'Occ-Net + explicit completion (Occ-Comp)' strategy.

low RfDNet to adopt a detection backbone, and use all points in the predicted bounding boxes to feed into the subsequent completion and reconstruction modules. Table 2 shows that it leads to an inferior performance compared to our complete pipeline. Figure 5 (b) shows that the pure-detection backbone tends to result in excessive false positive proposals, as mentioned by the authors of DIMR [16].

Effect of the disentanglement scheme: When removing 'Completion', it means that we remove the explicit completion module by directly connecting the decoder of the mesh reconstruction network after the instance encoder to obtain instance meshes from partial observations. As shown in Figure 5 (c), without completion, the network fails to learn reasonable shapes and leads to extremely low scores as shown in Table 2. We think the reason may be that we adopt no constraint or regularization technique over our Occ-Net latent space (such as the CVAE adopted by DIMR), making it hard for the network to learn directly from partial points. Introducing the explicit completion module solves this issue by providing the Occ-Net with complete points as input, and it requires no constraints on the latent space. More discussion can be found in Section 4.4.

Effect of each confidence score: In the bottom four rows of Table 2, 'Seg', 'Cls', 'Bbox', and 'Mesh' stands for the confidence scores of instance segmentation, semantic classification, bounding box regression and mesh prediction, respectively. Removing each one means to multiply the other three as the final confidence score. Clearly, compared to our full pipeline (the first row), removing each confidence score results in a worse performance.

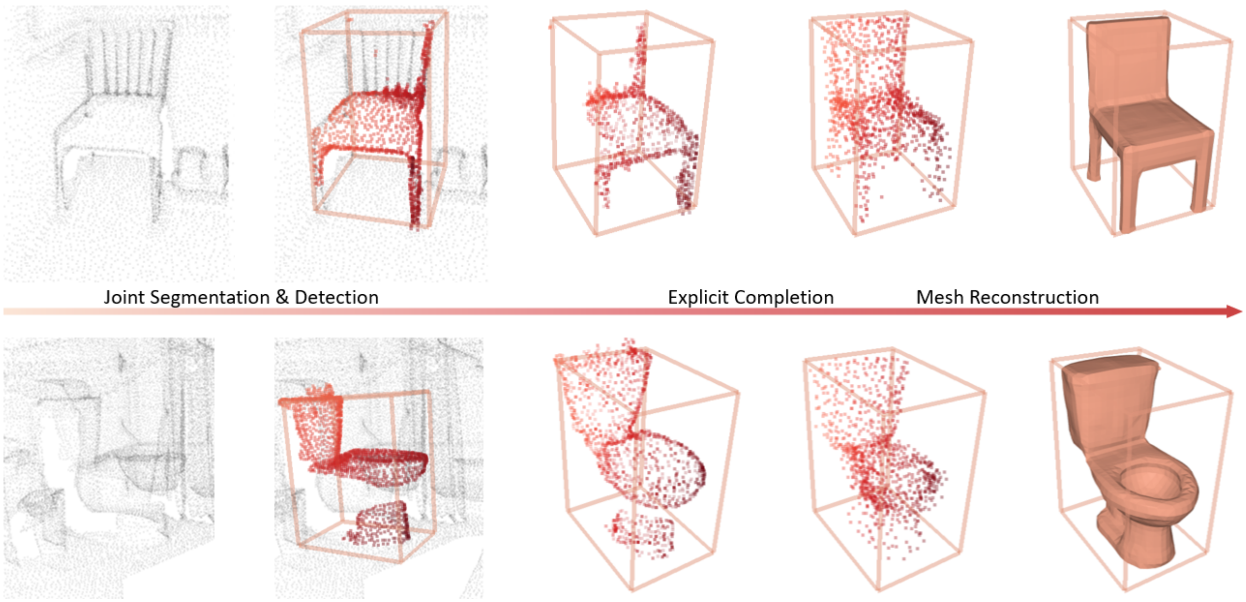

Fig. 8. Intermediate visual results of major steps in our pipeline. For clarity, we only visualize one instance in each scene.

TABLE 3
Using ground-truth segmentation and detection results significantly improves instance mesh reconstruction results.

	CD@0.1	CD@0.047
RfD-Net [15]	46.37	19.09
DIMR [16]	52.39	25.71
JIMR (ours)	50.67	26.14
GT_det_seg	**85.07**	**52.90**

TABLE 4
Comparisons of 3D detection and segmentation performance reported by mAP measured at bbox IoU and segmentation IoU with threshold 0.5. DIMR and JIMR detection results are evaluated on the full test set following RfD-Net, and the results of RfD-Net are from the authors.

	Detection		Segmentation
	stage	IoU@0.5	IoU@0.5
RfD-Net [15]	1-stage detection	35.10	-
DIMR [16]	2-stage detection	39.16	53.34
JIMR (ours)	1-stage detection	**40.19**	**61.73**

4.4 Analysis and Discussion

We conduct extra experiments to further analyze our JIMR and give some discussions.

Importance of high segmentation and detection accuracy. We design our framework based on the hypothesis that high segmentation and detection accuracy promote final reconstruction quality. To verify this, we conduct experiments by directly feeding ground truth segmentation and detection results to the subsequent completion and reconstruction modules. The results are reported in Table 3. We can see that using GT results indeed improves IMR performance by about 25% - 35%, thus proving our hypothesis.

Comparisons of detection and segmentation performance. As mentioned before, our proposed joint segmentation and detection backbone can make the two tasks promote each other. We now report the comparisons of bbox detection and instance segmentation in Table 4. Note that our network structure in the first stage is exactly the same as DIMR except that we further add a bbox regression head. Clearly, our method achieves higher accuracy in terms of both detection and segmentation within a single stage thanks to the joint learning strategy.

Discussion on explicit completion module. As shown in Table 2, when removing the completion module from our full pipeline, the reconstruction quality degrades dramatically; see the left part in Figure 6 (a) for the visual results. Clearly, compared to the ground truth mesh (right), when we remove completion, the network fails to converge to generate reasonable shapes.

Then, a natural question is: why can DIMR generate reasonable meshes with no explicit but only an implicit completion module? Actually, in DIMR, to facilitate the second-stage network to regress latent codes of a complete shape, latent space regularization is used to ensure that the latent codes produced by pre-trained mesh generator follow a standard normal distribution. Following this idea, we also tried latent regularization when training our pre-trained mesh generator. Yet, as shown in Figure 6 (b), though we have tried two different regularization techniques (i.e., L2 regularization and batch normalization), the associated performance is still worse than adopting no latent space constrain. This is consistent with the observation in [51] that a more regularized latent space comes at the cost of sacrificing performance to some degree. Therefore, we employ explicit completion instead of latent regularization.

Besides disentanglement of sub-tasks (completion and reconstruction) and high-quality mesh generation without sacrificing performance for latent regularization, the disentangled completion and reconstruction design has another advantage: enabling the usage of the pre-trained weights from existing powerful completion and reconstruction networks. As a result, reasonable meshes can be yielded even at

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3398737

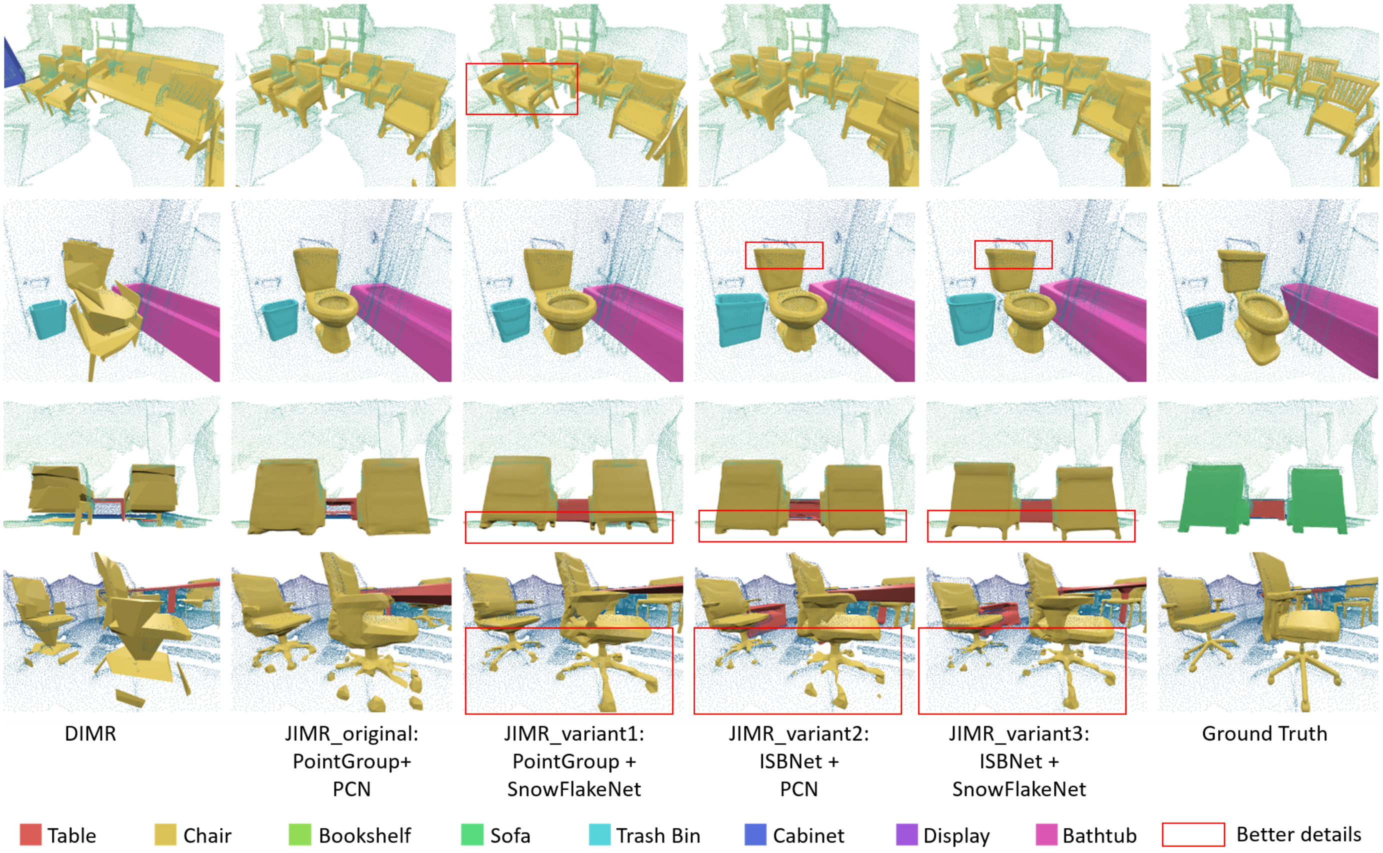
DIMR
JIMR_original: PointGroup+ PCN
JIMR_variant1: PointGroup + SnowFlakeNet
JIMR_variant2: ISBNet + PCN
JIMR_variant3: ISBNet + SnowFlakeNet
Ground Truth
Table
Chair
Bookshelf
Sofa
Trash Bin
Cabinet
Display
Bathtub
Better details

Fig. 9. Qualitative results of replacing our segmentation and completion modules from PointGroup and PCN to state-of-the-art ISBNet and SnowFlakeNet. The substitution slightly improved mesh reconstruction quality by capturing finer details.

the beginning of the second stage training, thus making the network training faster and easier. Specifically, we only train our JIMR for 4 epochs in the second stage that can already provide promising reconstruction results, as compared to DIMR with 256 epochs; see Figure 6 (c) for the reconstructed meshes in different early epochs.

To further evaluate the effect of our stage-2 strategy (i.e. Occ-Net with explicit completion), we adopt the same stage-1 network as our JIMR, and train a stage-2 network following DIMR (i.e. BSP-Net with latent regularization) till convergence with 300 epochs. Figure 7 demonstrates some qualitative comparisons using these two different stage-2 strategies. Clearly, our method yields better mesh reconstruction performance with only 4 epochs. Since BSP-Net represents 3D shapes by planes and lacks the ability to precisely model curved surfaces, it often produces meshes with a low-poly style or uneven artifacts, and fails to reconstruct complicated shapes like the office chair.

Visual results of segmentation, detection, completion & reconstruction. Figure 8 shows the intermediate visual results of major steps in our pipeline, including instance segmentation, object detection, point cloud completion and the final mesh reconstruction. For clarity, we only visualize one instance in each scene. Generally, given a successfully segmented instance, our method can predict a complete bounding box. The completion network can also recover missing regions, thus facilitating a high-quality reconstructed instance mesh.

TABLE 5
Quantitative results of module substitution experiments on the full test set of ScanNetV2 by replacing our segmentation and completion modules from PointGroup and PCN to state-of-the-art ISBNet and SnowFlakeNet. The substitution further enhances the performance.

Method	Segmentation Module	Completion Module	CD@0.1	CD@0.047
JIMR_original	PointGroup [4]	PCN [17]	50.67	26.14
JIMR_variant1	PointGroup [4]	SnowFlakeNet [35]	50.83	27.12
JIMR_variant2	ISBNet [3]	PCN [17]	51.33	30.76
JIMR_variant3	ISBNet [3]	SnowFlakeNet [35]	**53.41**	**33.02**

Module substitution analysis. In our initial JIMR implementation, we adopted the same segmentation module PointGroup [4] as in DIMR for a fair comparison, and we adopted PCN [17] as the completion module by considering the computation efficiency. We argue that our main contribution is a unified framework where the sub-modules can be replaced by existing ones. Hence, we conducted extra experiments by replacing our segmentation and completion modules with existing SOTA ones. By considering the ease of implementation and the network performance, we here choose ISBNet [3] for instance segmentation and SnowFlakeNet [35] for shape completion. Specifically, when adopting SnowFlakeNet, we directly used it to replace PCN. When adopting ISBNet, we added an oriented bounding box head to its point-wise prediction module, like we did to PointGroup as in JIMR. Table 5 shows the quantitative results, where JIMR variants with stronger sub-modules outperform

TABLE 6
Run time analysis. Our method is the fastest with a medium number of network parameters.

	Avg. Time Per Scene (s)	Parameters
RfD-Net [15]	3.308	**15.23**M
DIMR [16]	15.320	53.51M
JIMR (Ours)	**2.074**	33.17M

our original implementation. Figure 9 shows the visual results. In general, we have the following observations by viewing the reconstructed meshes of these methods.

- *Firstly*, our JIMR and its variants with SOTA sub-modules exhibit a competitive visual effect compared to DIMR.
- *Secondly*, the three JIMR variants with SOTA sub-modules yield sometimes, though not always, finer details than the original JIMR implementation, such as the hollowed-out armrest, toilet top, small sofa legs, office chair legs, etc., as marked by red boxes in Figure 9. We believe that stronger sub-modules promote the network's ability to capture fine details when reconstructing meshes.
- *Lastly*, all JIMR variants share the same "mesh style" as JIMR, compared to the low-poly style of DIMR. This is natural since they only differ in detailed implementation of sub-modules, but share the same overall framework and mesh reconstruction module.

Run time analysis. Table 6 shows the average inference time per scene and the number of parameters of RfD-Net [15], DIMR [16] and our JIMR. We run the inference code on the same test set for all the three methods and calculate the average time. Our method is the fastest and RfD-Net is slightly slower than ours, both of which are significantly faster than DIMR since the Marching Cubes algorithm of Occ-Net in RfD-Net and our JIMR with resolution 32^3 is much faster than Constructive Solid Geometry method of BSP-Net in DIMR. We have the medium parameter number compared with DIMR and RfD-Net, since we have a voxel-based backbone as in DIMR, and a point-based second-stage network as in RfD-Net.

5 Conclusion

We propose JIMR, a joint framework for point-based indoor scene instance mesh reconstruction, where sub-modules can be flexibly substituted. It adopts a two-stage pipeline for both semantic and geometry learning. The first stage is a joint segmentation and detection network to produce proposal instance point clouds and 3D bounding boxes together, which are used in the second stage to complete the point clouds and then reconstruct instance meshes. Experiments show that our method improves mesh reconstruction quality by a significant margin compared to state-of-the-art approaches, especially regarding the visual effect. Failure case analysis is provided in the supplementary file. In the future, we plan to investigate the possibility of adopting generative models to enhance the local details of produced instance meshes. Further, we may also consider exploring 2D supervision by rendering 3D instances into 2D images for better mesh reconstruction.

References

[1] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D semantic segmentation with superpoint transformer," in *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[2] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3D: Mask Transformer for 3D Semantic Instance Segmentation," in *International Conference on Robotics and Automation (ICRA)*, 2023.

[3] K. N. Tuan Duc Ngo, Binh-Son Hua, "ISBNet: a 3D Point Cloud Instance Segmentation Network with Instance-aware Sampling and Box-aware Dynamic Convolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4867–4876.

[5] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical aggregation for 3D instance segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 467–15 476.

[6] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo, "Soft-Group for 3D instance segmentation on 3D point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2569–2578.

[8] J. Hou, A. Dai, and M. Nießner, "RevealNet: Seeing behind objects in RGB-D scans," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2098–2107.

[9] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1746–1754.

[10] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4578–4587.

[11] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 523–540.

[12] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "Sa-ConvOnet: Sign-agnostic optimization of convolutional occupancy networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 6504–6513.

[13] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "NeuralRecon: Real-time coherent 3D reconstruction from monocular video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 598–15 607.

[14] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6290–6301.

[15] Y. Nie, J. Hou, X. Han, and M. Nießner, "RfD-Net: Point scene understanding by semantic instance reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4608–4618.

[16] J. Tang, X. Chen, J. Wang, and G. Zeng, "Point scene understanding via disentangled instance mesh reconstruction," in *European Conference on Computer Vision (ECCV)*, 2022.

[17] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[18] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D reconstruction in function space," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4460–4470.

[19] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9277–9286.

[20] Z. Chen, A. Tagliasacchi, and H. Zhang, "BSP-Net: Generating compact meshes via binary space partitioning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 45–54.

[21] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4421–4430.

[22] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3D instance segmentation on point clouds," *Advances in neural information processing systems*, vol. 32, 2019.

[23] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with Pointformer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7463–7472.

[25] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] D. Zhou, J. Fang, X. Song, L. Liu, J. Yin, Y. Dai, H. Li, and R. Yang, "Joint 3D instance segmentation and object detection for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1836–1846.

[27] O. Unal, L. Van Gool, and D. Dai, "Improving point cloud semantic segmentation by learning 3D object detection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 2950–2959.

[28] Y. Chen, H. Li, R. Gao, and D. Zhao, "Boost 3D object detection via point clouds segmentation and fused 3D GIoU L1 loss," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[29] Y. Zhong, M. Zhu, and H. Peng, "VIN: Voxel-based implicit network for joint 3D object detection and segmentation for lidars," in *British Machine Vision Conference (BMVC)*, 2021.

[30] D. Stutz and A. Geiger, "Learning 3D shape completion from laser scan data with weak supervision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5868–5877.

[32] Y. Nie, Y. Lin, X. Han, S. Guo, J. Chang, S. Cui, J. Zhang *et al.*, "Skeleton-bridged point completion: From global inference to local adjustment," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 119–16 130, 2020.

[33] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[34] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *European Conference on Computer Vision (ECCV)*, 2022.

[35] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. Han, "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[36] Z. Chen, F. Long, Z. Qiu, T. Yao, W. Zhou, J. Luo, and T. Mei, "Anchorformer: Point cloud completion from discriminative nodes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 581–13 590.

[37] S. Li, P. Gao, X. Tan, and M. Wei, "Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 9466–9475.

[38] C. Xie, C. Wang, B. Zhang, H. Yang, D. Chen, and F. Wen, "Style-based point generator with adversarial rendering for point cloud completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4619–4628.

[39] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7662–7670.

[40] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[42] P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, "AutoSDF: Shape priors for 3D completion, reconstruction and generation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 306–315.

[43] X. Yan, L. Lin, N. J. Mitra, D. Lischinski, D. Cohen-Or, and H. Huang, "ShapeFormer: Transformer-based shape completion via sparse representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6239–6249.

[44] W. E. Lorensen and H. E. Cline, "Marching Cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[45] D. Rukhovich, A. Vorontsova, and A. Konushin, "FCAF3D: Fully convolutional anchor-free 3D object detection," in *European Conference on Computer Vision (ECCV)*, 2022.

[46] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[47] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5939–5948.

[48] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.

[49] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "ShapeNet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[50] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2CAD: Learning CAD model alignment in RGB-D scans," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2614–2623.

[51] R. Morrow and W.-C. Chiu, "Benefiting deep latent variable models via learning the prior and removing latent regularization," *arXiv preprint arXiv:2007.03640*, 2020.

Qiao Yu received the bachelor's degree from Huazhong University of Science and Technology, China, in 2019. She is currently pursuing the Ph.D. degree with the Embedded and Pervasive Computing (EPIC) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology. Her research interest includes deep learning, 3D vision and scene understanding.

Xianzhi Li received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong. She is currently an Associate Professor at the Huazhong University of Science and Technology. Prior to that, she was a postdoctoral fellow at The Chinese University of Hong Kong. Her research interests include 3D vision, computer graphics, and deep learning. She serves as the reviewer of several conferences and journals, including TVCG, CVPR, ICCV, etc.

Yuan Tang received the bachelor's degree in College of Information and Software Engineering from University of Electronic Science and Technology of China (UESTC) in 2019. Currently, he is a Ph.D. candidate of Embedded and Pervasive Computing (EPIC) Lab in School of Computer Science and Technology at Huazhong University of Science and Technology (HUST). His research interests include 3D representation learning, self-supervised learning, etc.

Jinfeng Xu received the bachelor's degree from the College of Control Science and Engineering, Shandong University (SDU), China, in 2020. He is currently pursuing the Ph.D. degree with the Embedded and Pervasive Computing (EPIC) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology. His research interest includes deep learning, 3D vision and scene understanding.

Long Hu has been an assistant professor in the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), China, since 2017. He was a visiting student in the Department of Electrical and Computer Engineering, University of British Columbia from August. 2015 to April 2017. His research includes the Internet of Things, software defined networking, caching, 5G, body area networks, body sensor networks, and mobile cloud computing.

Yixue Hao received the Ph.D. degree in computer science from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, HUST. His current research interests include 5G networks, the Internet of Things, edge computing, edge caching, and cognitive computing.

Min Chen has been a Full Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), since February 2012. He is currently the Director of the Embedded and Pervasive Computing (EPIC) Laboratory, and the Director of the Data Engineering Institute, HUST. His Google Scholar citations reached more than 33,100 with an H-index of 87 and i10-index of 261. His research interests include cognitive computing, 5G networks, wearable computing, big data analytics, robotics, machine learning, deep learning, emotion detection, and mobile edge computing. He received the IEEE Communications Society Fred W. Ellersick Prize in 2017 and the IEEE Jack Neubauer Memorial Award in 2019. He is the Chair of the IEEE GLOBECOM 2022 eHealth Symposium. He is the Founding Chair of the IEEE Computer Society Special Technical Communities on Big Data.