ME-PCN: Point Completion Conditioned on Mask Emptiness

Bingchen Gong[1], Yinyu Nie[2], Yiqun Lin[3], Xiaoguang Han*[3], and Yizhou Yu*[1]
[1]The University of Hong Kong, [2]Technical University of Munich, [3]SSE, CUHK(SZ)
bccs@connect.hku.hk, yinyu.nie@tum.de, lyq211003@gmail.com, hanxiaoguang@cuhk.edu.cn, yizhouy@acm.org

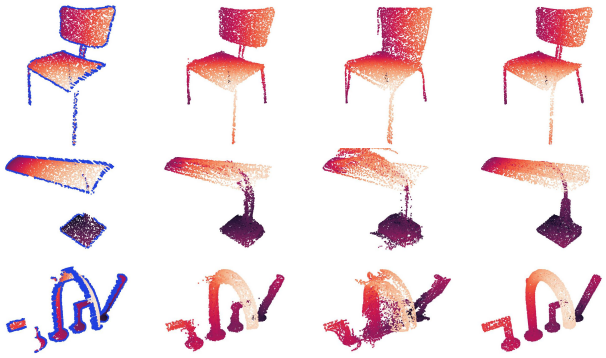
Abstract

Point completion refers to completing the missing geometries of an object from incomplete observations. Mainstream methods predict the missing shapes by decoding a global feature learned from the input point cloud, which often leads to deficient results in preserving topology consistency and surface details. In this work, we present ME-PCN, a point completion network that leverages ***emptiness*** *in 3D shape space. Given a single depth scan, previous methods often encode the occupied partial shapes while ignoring the empty regions (e.g. holes) in depth maps. In contrast, we argue that these 'emptiness' clues indicate shape boundaries that can be used to improve topology representation and detail granularity on surfaces. Specifically, our ME-PCN encodes both the occupied point cloud and the neighboring 'empty points'. It estimates coarse-grained but complete and reasonable surface points in the first stage, followed by a refinement stage to produce fine-grained surface details. Comprehensive experiments verify that our ME-PCN presents better qualitative and quantitative performance against the state-of-the-art. Besides, we further prove that our 'emptiness' design is lightweight and easy to embed in existing methods, which shows consistent effectiveness in improving the CD and EMD scores.*

1. Introduction

Capturing 3D data for objects around us is as easy as taking a picture with cell phones thanks to the popularity of common 3D scanning sensors like LIDAR and depth cameras. Such availability has greatly enriched the practical applications in vision and robotics communities.

Different from image sensors, data from 3D scanners usually come incompletely with a much lower resolution, e.g., depth maps with missing values. To recover complete shapes from the partial inputs, volumetric and view-based projection methods leverage 3D convolutions to represent shapes into voxel grids. However, those 3D convolutions, suffering from expensive memory and computational cost, are bottlenecked by the resolution-computation balance. It becomes even disadvantaged when the inputs are unordered and sparse. Implicit methods learn a signed distance function (SDF) to represent shape surfaces and are capable of reaching any high resolution. However, they still rely on voxel grids, and an extra computation-intensive post-processing step is required to extract the final shape surface. In contrast, the point cloud is a more compact representation of 3D shapes. Compared with voxels, it is more scalable and computation-efficient to express shapes with different granularity. Several deep neural networks have been proposed to directly take advantage of point cloud representation, such as PCN [31], TopNet [21], MSN [10].



(a) Input (b) Ours (c) MSN (d) GT

Figure 1: Given a partial scan (a), our method encodes the spatial emptiness (blue points in (a)) neighboring to observations and predicts complete and topology consistent surfaces (b) compared with MSN [10] on the ground-truth (d).

A key concept in existing point completion methods is an encoder-decoder architecture followed by an optional refining process, where the completed point cloud is generated from an encoded global feature. The widely used encoders for point cloud are PointNet [15] and its variant PointNet++ [16]. With the encoder-decoder design, current methods directly predict the complete points from the visible, occupied input points, while ignoring the unoccupied,

*Corresponding author

empty regions in the inputs. In our view, the unoccupied regions are the complement of shape occupancy, thus also indicating the topology of 3D objects. Compared with learning from observable shapes, learning the emptiness presents extra significance, especially for complex shapes, such as non-convex surfaces with holes. In other words, the input scan tells not only the shape occupancy but also 'where should not be occupied'. However, current methods predict shapes in the whole 3D space, which could be insensitive to subtle topologies. In our method, the emptiness in the input can inform our network '*which regions should* ***not*** *be occupied*', and helps to keep consistent shape topology. Such emptiness information can be encoded by a mask given a viewpoint and can be easily obtained by thresholding the input depth scan or extracted from RGB images.

Inspired by the above, we introduce ME-PCN, a novel point completion network informed with mask emptiness. To encode the emptiness clues on a mask, 3D rays are radiated from the viewpoint towards the empty regions of the mask. All points along the rays will be encoded as empty points. In ME-PCN, only empty points that are in the neighborhood of visible points are processed by neural networks in addition to the input point cloud. Since visible points and empty points have totally different semantics, two separate networks are used to encode them into two global features.

For surface completion, directly decoding shapes from global features can predict plausible structures but usually results in coarse and over-smoothed surfaces [10, 31]. It also neglects subtle structures on the boundary between real and empty points. To this end, a final refining stage is performed after the coarse decoder. Local features are learned from neighboring empty and visible points for each coarse point, which augments the coarse input with on-surface details. The effectiveness of the emptiness inputs is verified both qualitatively and quantitatively. In summary, the main contributions of our work are:

- We provide a novel encoding modality for point completion. Prior arts learn complete shapes only from visible points, while our method involves *emptiness* learning to represent consistent shape topology and improves surface details.
- We propose ME-PCN to learn the shape emptiness from depth masks. Given a depth scan, 3D rays radiated from the viewpoint to empty regions on the mask are encoded to represent the emptiness in 3D shape space. It informs ME-PCN of the shape boundaries and improves the completion performance.
- Extensive experiments verify that our emptiness learning strategy can be easily embedded into modern point-based shape completion pipelines to improve the CD and EMD scores, which further makes our method outperform the state-of-the-art.

2. Related Work

Shape Representation and Reconstruction Given an object observation (e.g., images, depth maps, point clouds), shape reconstruction aims to predict a plausible geometry and recover the shape surfaces. Early works extend the advantages of 2D convolutions in image perception to 3D, and adopt 3D convolutions to reconstruct shapes with discretized voxels [5, 2, 3, 7, 19, 18]. These works pioneered the 3D shape analysis modality but the expensive 3D convolutions make them bottlenecked by the resolution-efficiency problem, demanding an extra Octree to improve local details [17, 20, 23]. On this top, some works represent shapes with SDFs [11, 1, 9], which theoretically can achieve any high resolution. However, the SDF methods still rely on voxel grids and require time-consuming post-processing to extract mesh surfaces. Besides, both the voxel and SDF methods can not well express shape boundaries thus leading to inferior surface details. Some other works directly generate surface meshes as the output [22, 6]. These methods approximate the target surfaces by deforming template meshes (e.g., planes or spheres), where the shape topology is usually restrained by the original templates. To this end, other works [14, 12] learn to modify the topology of template meshes, but it requires massive computations and often results in open boundaries. Besides, the kernel in the above methods is an encoder-decoder structure, where shapes are decoded or deformed from a global feature, decoding from which would be insensitive to boundary details and produce over-smoothed results. In our method, we leverage the emptiness information close to the shape boundary, and demonstrate its effectiveness in improving surface details.

Shape Completion Different from shape reconstruction, shape completion focuses on predicting a complete shape from a partial, observable surface. Similar to reconstruction methods, many works also adopt an encoder-decoder structure backboned by 3D CNNs or MLPs [5, 3, 4, 29] to represent shapes with voxels or points. Since such a structure cannot produce fine details, [31, 10, 28] adopt a coarse-to-fine completion strategy to firstly predict coarse points with MLPs and subsequently generate dense and refined results. Besides, [24, 26] propose skip-connections or cascaded blocks to revisit shallow-end point features to complement surface details. [8] provides a multi-resolution encoder to perceive shape details under different granularity, and deploys a pyramid network to recover complete points by increasing the resolution. However, the key concept in these methods is how to improve point features to encode and decode more enriched shape signals. There are no explicit constraints to keep topology consistency with the target shape. On this point, [13] provides a skeleton-bridged method to predict surface points by first learning shape skeletons. However, the skeletal points of objects

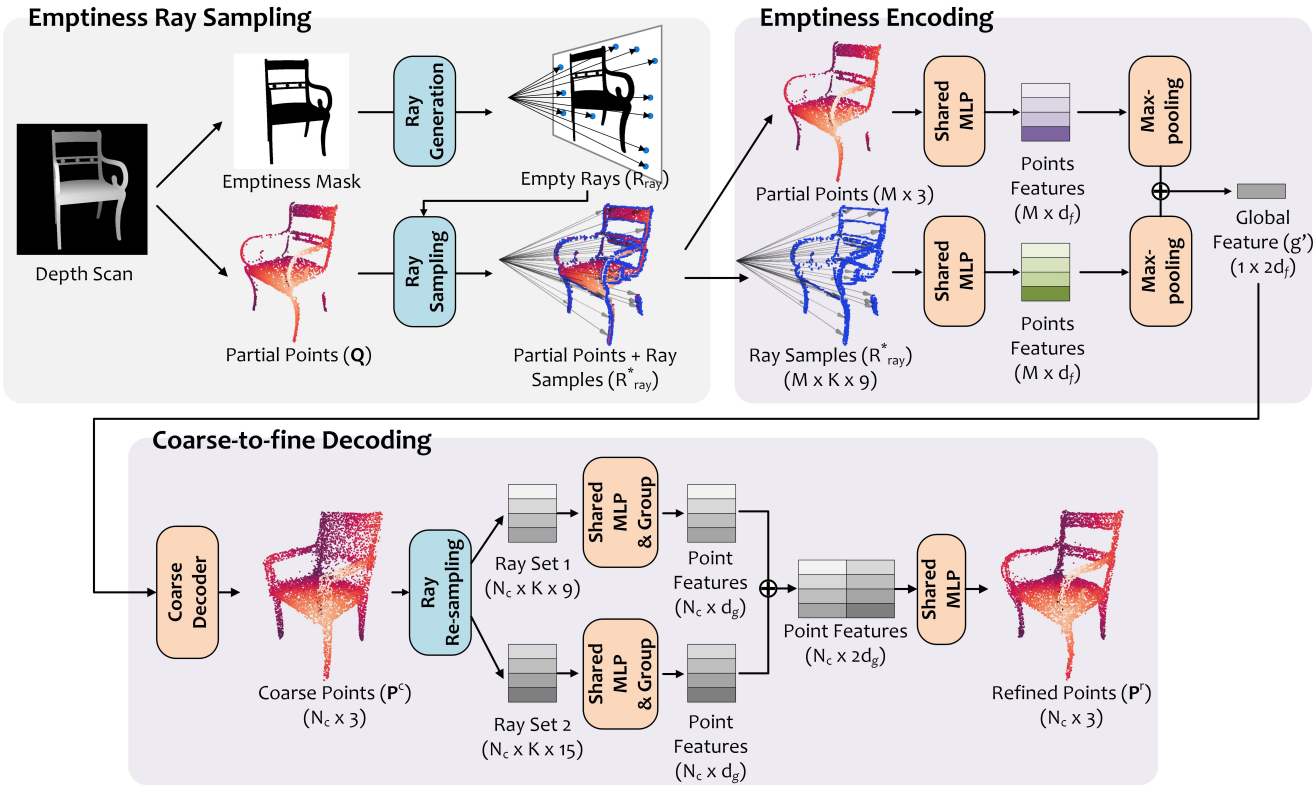


Figure 2: Architecture of ME-PCN. Given a partial scan, our network learns the spatial occupancy and emptiness from the observable points $\mathbf{Q}$ and 3D empty rays $\mathcal{R}_{ray}$ from the object mask. After sampling the neighboring empty rays $\mathcal{R}^*_{ray}$ queried by $\mathbf{Q}$, an *Emptiness Encoder* is adopted to learn a global shape feature g' by encoding the shape occupancy $\mathbf{Q}$ and spatial emptiness $\mathcal{R}^*_{ray}$ separately. We firstly predict a coarse shape $\mathbf{P}^c$ to obtain a rough structure. To recover surface details. A *Ray Re-sampling* strategy is adopted to obtain two sets of empty rays from $\mathcal{R}_{ray}$ and $\mathcal{R}^*_{ray}$ respectively queried by $\mathbf{P}_c$. Two separate MLPs are used to respectively learn the point features before concatenation to predict the refined points $\mathbf{P}^r$.

are extremely sparse. Any skeletal errors would directly influence the structure and surface quality. Besides, [28] involves an extra adversarial point rendering by minimizing the depth map distance with the ground-truth under different views. In our work, we provide a lightweight approach to encode topological information by learning the 'empty points' close to the observed input points. It informs our network that which regions are unoccupied thus helps to predict consistent sub-structures.

3. Approach

Our method consumes unordered ray sets of points as the input. A ray set is denoted as a set of 3D vector pairs $\mathcal{R}_{ray} = \{(p_i, v_i)|i = 1, ..., N\}$, where each start point $p_i \in \mathbf{R}^3$ is a vector of 3D coordinates. $v_i \in \mathbf{R}^3$ is the normalized vector of its ray direction from the viewpoint to p_i.

3.1. Ray Generation from Masks

Our method is illustrated in Figure 2. Given a depth map with the corresponding viewpoint, we define a 2D 'emptiness mask' as the following:

$$mask_{s,t} = \begin{cases} 1, & \text{if position } s,t \text{ is empty,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A mask can be easily obtained by thresholding the depth map or extracted from the corresponding RGB image. Once we have the $mask$, an empty ray set $\mathcal{R}_{ray}$ can be calculated by back-projecting the $mask$ into 3D space:

$$p_i = \textit{Back-project}(s, t, d_{far}) \quad (2)$$

$$l \cdot v_i = p_i - \textit{Back-project}(s, t, d_{near}) \quad (3)$$

Back-project is the reverse process of projection, which lifts a 2D image or depth map back to 3D space. Position (s, t) is the i-th non-zero elements in $mask$, which indicates an empty pixel by Equation 1. l is the normalization factor to ensure v_i is a unit vector. d_{far} and d_{near} are the depth values of the farest plane and the nearest plane, respectively. Therefore, we let $\mathcal{R}_{ray}$ to represent the 3D empty points back-projected from $mask$. d_{far} and d_{near} are kept consistent among all shapes.

3.2. Ray Points Sampling

Emptiness only has meanings when there are subjects in its neighborhood. It means that both the empty rays $\mathcal{R}_{ray}$ and visible points are not isolated. Our network should learn the local structures from the nearby empty points and the combinatorial interactions among local structures.

However, in $\mathcal{R}_{ray}$, many rays are actually too far from the subject and therefore convey little information. In this section, we provide a sampling strategy to obtain informative rays from $\mathcal{R}_{ray}$ as the input for ME-PCN. Those rays should be close to shape surfaces to convey local details. Specifically, given a real point $q_j \in \mathbf{R}^3$ on the shape surface, we sample a subset of rays from $\mathcal{R}_{ray}$ in the neighborhood of q_j, where we choose K nearest rays for each q_j. On each neighboring ray $\mathbf{r}_k \in \mathcal{R}_{ray}$, we select the nearest point as an empty point candidate $p_k^e \in \mathbf{R}^3$. Thus each real point q_j has K candidates of empty points $\{p_k^e\}$, $k = 1, 2, ..., K$. The Euclidean distance $\|D_{\mathbf{r},q}\|$ between a ray $\mathbf{r} = (p, v) \in \mathcal{R}_{ray}$ and a real point q is defined by:

$$p^e = p - [(p - q) \cdot v]\, v, \tag{4}$$

$$D_{\mathbf{r},q} = p^e - q, \tag{5}$$

where p^e is the nearest point from ray $\mathbf{r}$ to real point q. $D_{\mathbf{r},q} \in \mathbf{R}^3$ is the offset vector from q to p^e.

After sampling, for each real point q, we combine its K nearest empty points $\{p_k^e\} \in \mathbf{R}^{K,3}$ with the corresponding ray direction vectors $\{v_k\} \in \mathbf{R}^{K,3}$ and the offset vectors $\{D_k\} \in \mathbf{R}^{K,3}$. Denote that there are M visible points $\mathbf{Q} \in \mathbf{R}^{M,3}$, then we input our network with the processed rays

$$\mathcal{R}^*_{ray} = \{\{p_k^e\}, \{D_k\}, \{v_k\}\} \in \mathbf{R}^{M \times K \times 9}. \tag{6}$$

Both p_k^e and D_k indicate the spatial neighborhood to real point q in Euclidean space. This could provide explicit cues for our network to capture local structures from nearby rays.

3.3. Emptiness Encoding

We illustrate the architecture in Figure 2. Our approach takes a partial point cloud $\mathbf{Q}$ and sampled rays $\mathcal{R}^*_{ray}$ as inputs and encodes them into a global feature vector (GFV) with emptiness semantics, which will be used to predict complete point cloud with a coarse-to-fine strategy.

In the encoding stage, since visible points $\mathbf{Q}$ and empty points in $\mathcal{R}^*_{ray}$ have totally different semantics with non-identical scale, we use two separate networks to encode them into two global feature vectors, respectively.

The encoder part consists of two Feature Encoding (FE) layers to respectively process visible points $\mathbf{Q}$ and sampled rays $\mathcal{R}^*_{ray}$. The first FE layer consumes the coordinates of visible points $\mathbf{Q}$ as the input. A shared multi-layer perceptron (MLP) consisting of two linear layers with ReLU activation is used to transform points $\{q_i | q_i \in \mathbf{Q}\}$ into point features $\{f_i\} \in \mathbf{R}^{M,d_f}$. The second FE layer takes the sampled rays $\mathcal{R}^*_{ray}$ as the input and produces point features $\{g_i\} \in \mathbf{R}^{N,d_f}$, similar to the first FE layer.

The two FE layers output two feature matrices $F = \{f_i\}$, $G = \{g_i\}$. A point-wise max-pooling is respectively performed on F, G to obtain d_f-dimensional global features f and g. Lastly, f and g are concatenated together to form a single global feature vector $g' = [f, g] \in \mathbf{R}^{2d_f}$.

3.4. Coarse-to-Fine Decoding by Ray Resampling

From the global feature g', we can directly decode the complete point cloud that captures the overall shape following [31, 10]. However, as discussed in Section 2, decoding shapes merely from a global feature would neglect local details and result in over-smoothed structures. To this end, a refining stage that operates on generated coarse points is usually preferred. In this part, we build our decoder with a coarse-to-fine strategy. A coarse decoder from [10] is adopted to firstly predict a coarse-grained but structure completed points $\mathbf{P}^c \in \mathbf{R}^{N_c \times 3}$. N_c is the number of coarse points. However, unlike [31, 24] where the local features are complemented by skip-connecting the shallow-end layer responses, we provide an explicit approach to revisit the emptiness information and decode fine-grained surface points. Our design is based on insights: 1) coarse points decoded from a global feature are still not accurate to preserve a consistent boundary compared to the ground-truth due to its roughness; 2) the point information in $\mathcal{R}^*_{ray}$ conveys the surface clues that can improve shape detail recovery.

For the first part, we inform our decoder with the emptiness information (i.e., 'emptiness mask' in Equation 1). It tells our decoder '*whether the coarse points are in empty regions*'. For the second part, we inform our decoder with the shape information. It tells our decoder *'what the real surface looks like'*. To realize these, given the coarse points set $\mathbf{P}^c$, we respectively resample two sets of empty rays $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$ as the input for the surface refining decoder, which is illustrated in Figure 3.

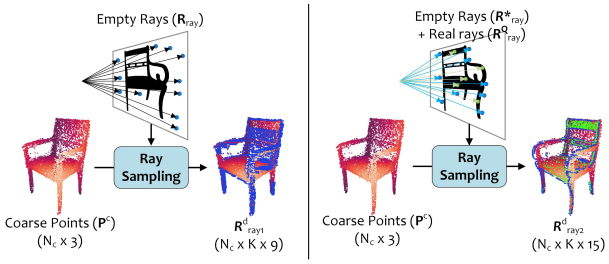


Figure 3: Resampling rays queried by coarse points $\mathbf{P}^c$ into $\mathcal{R}^d_{ray1}$ (left) and $\mathcal{R}^d_{ray2}$ (right). Blue points denote the sampled rays from empty rays $\mathcal{R}_{ray}$ (left) or $\mathcal{R}^*_{ray}$ (right). Green points represent the rays sampled from visible points. Blue points imply whether coarse points are in empty regions. Green points reveal the position of visible points.

Sampling $\mathcal{R}^d_{ray1}$: We sample $\mathcal{R}^d_{ray1}$ with the same method of sampling $\mathcal{R}^*_{ray}$ (see Section 3.2). The only difference is that the visible points $\mathbf{Q}$ in Section 3.2 is replaced with the coarse points $\mathbf{P}^c$. Then the sampled empty rays $\mathcal{R}^d_{ray1} \in \mathbf{R}^{N_c \times K \times 9}$ will tell whether a coarse point in $\mathbf{P}^c$ is in an 'empty' region.

Sampling $\mathcal{R}^d_{ray2}$: For $\mathcal{R}^d_{ray2}$, we adopt the same sampling method and use coarse points $\mathbf{P}^c$ as the query to collect neighboring rays into $\mathcal{R}^d_{ray2}$. The difference is that, we no longer sample rays from $\mathcal{R}_{ray}$ but from the union of its subset $\mathcal{R}^*_{ray}$ and the ray set $\mathcal{R}^Q_{ray}$ (generated from visible points $\mathbf{Q}$). $\mathcal{R}^*_{ray}$ presents the boundary rays in $\mathcal{R}_{ray}$ (see Section 3.2). With the definition in Equation 6, the rays generated from $\mathbf{Q}$ are defined by:

$$\mathcal{R}^Q_{ray} = \{(q_i, \vec{0}, v_i) | i = 1, ..., M\} \tag{7}$$

$$l \cdot v_i = q_i - \textit{Back-project}(s, t, d_{near}), \tag{8}$$

where q_i is the 3D coordinates of a real point in $\mathbf{Q}$. (s, t) is its projection on the depth map. v_i is the unit directional vector from camera viewpoint to q_i. Since q_i is a real point, so the offset between ray (q_i, v_i) and a real point is $\vec{0}$ (see Equation 5). l is a factor to ensure v_i is a unit vector. Note that to satisfy the assumption: all points on rays are empty points. We only consider those rays $\{(q_i, \vec{0}, v_i)\} \subset \mathcal{R}^Q_{ray}$ whose depth values at $\{(s, t)\}$ is larger than the corresponding depths of coarse points $\mathbf{P}^c$ on the image plane.

Note that we sample $\mathcal{R}^d_{ray2}$ from $\{\mathcal{R}^*_{ray}, \mathcal{R}^Q_{ray}\}$. Following Section 3.2, for each point $p^c \in \mathbf{P}^c$ we sample K nearest rays and concatenate the empty points $\{p^{c,e}_k\}$, offset vectors $\{D^c_k\}$ with the corresponding empty ray informtion $\{\mathbf{r}^c\} \subset \mathcal{R}^*_{ray} \cup \mathcal{R}^Q_{ray}$. Therefore, $\mathcal{R}^d_{ray2} = \{\{p^{c,e}_k\}, \{D^c_k\}, \{\mathbf{r}^c\}\} \in \mathbf{R}^{N_c \times K \times 15}$

Decoding Refined Shape: Rays in $\mathcal{R}^d_{ray1}$ represent empty space neighboring to coarse points, while $\mathcal{R}^d_{ray2}$ informs the coarse points with the real shape boundary. Two FE layers are respectively used to encode $\mathcal{R}^d_{ray1}$ and $\mathcal{R}^d_{ray2}$. For $\mathcal{R}^d_{ray1}$, a shared MLP consisting of two linear layers with ReLU activation are used to transform points in $\mathcal{R}^d_{ray1} \in \mathbf{R}^{N_c \times K \times 9}$ into a grouped point feature vector $\{f^d_i\} \in \mathbf{R}^{N_c \times d_g}$. The second FE layer consumes $\mathcal{R}^d_{ray2}$ and produce a grouped point feature vector $\{g^d_i\} \in \mathbf{R}^{N_c \times d_g}$, similar to the first FE layer. The grouping operation is shown as ($\{f^d_i\}$ is taken as an example):

$$f^d_i = \sum_{k=1}^{K} \sum_{d=1}^{d_g} w(k, d) \mathbf{r}(i, k, d), \mathbf{r} \in \mathcal{R}^d_{ray1}, \tag{9}$$

where $\{w\}$ are the weights calculated following [27]. We concatenate $\{f^d_i\}$ and $\{g^d_i\}$ to regress the coordinates of complete surface points as the refined output $\mathbf{P}^r$.

3.5. Loss Function

We design our loss function via the Earth Mover's Distance (EMD) and the regularizer $\mathcal{L}_{expansion}$ from [10]:

$$\begin{aligned} \mathcal{L} =& \text{EMD}(\mathbf{P}^c, \mathbf{P}^{gt}) + \lambda_1 \cdot \mathcal{L}_{expansion} \\ &+ \lambda_2 \cdot \text{EMD}(\mathbf{P}^r, \mathbf{P}^{gt}). \end{aligned} \tag{10}$$

The EMD measures the distance from the coarse prediction $\mathbf{P}^c$ (or the refined prediction $\mathbf{P}^r$) to the ground-truth surface points $\mathbf{P}^{gt}$. The regularizer ensures point patches in $\mathbf{P}^c$ fit in local areas and not overlap too much. λ_1, λ_2 are two weights of importance that balance different losses.

4. Experiment and Evaluation

Network Specifications. Our network architecture is illustrated in Figure 2, where the coarse decoder is adopted following [10]. The parameters and layer information in Section 3 are detailed in the supp. material.

Dataset. We evaluate our methods on a subset of the ShapeNet dataset. 14 categories that contain a large number of models are selected where 29,795 CAD models are included. We report our evaluation on six categories: faucet, cabinet, table, chair, vase and lamp. The others are included in the supp. material. The complete point clouds are created by uniformly sampling $n_{gt} = 8192$ points on the mesh surfaces and the partial point clouds are generated by back-projecting 2.5D depth images into 3D. All CAD models are normalized into $[-1, 1]^3$ and located at the origin. For each category, we sample 9,000 pairs of partial and complete point clouds from different models using random viewpoint, resulting in 9,000×14 pairs of point clouds, where 10% of them for testing, and the rest are for training.

Benchmark. To validate our performance, we extensively compare our method with the state-of-the-art including PCN [31], PF-Net [8], P2P-Net [30], SoftPoolNet[25], CRN [24], GR-Net [29], MSN [10] and SK-PCN [13]. All methods are inputted with 5,000 points. Two resolutions of output points (2048 and 8192) are compared considering some methods support upsampling while the others do not. Besides, we also embed our emptiness learning into PCN (i.e. PCN+ray) and MSN (i.e. MSN+ray) to verify its effectiveness to different backbones. The network details of PCN+ray are illustrated in the supp. material.

Model Training. We trained all models on 2×NVIDIA GeForce RTX 2080 Ti GPUs for 25 epochs with a batch size of 16. The initial learning rate is 10^{-3} and is decayed by 0.1 per 10 epochs. Adam is used as the optimizer.

4.1. Comparison with Existing Methods

We list the qualitative results in Figure 4. For quantitative evaluation, since some methods (PCN [31], CRN [24], GR-Net [29], and MSN [10]) support upsampling to recover higher resolution of outputs, we compare our methods with

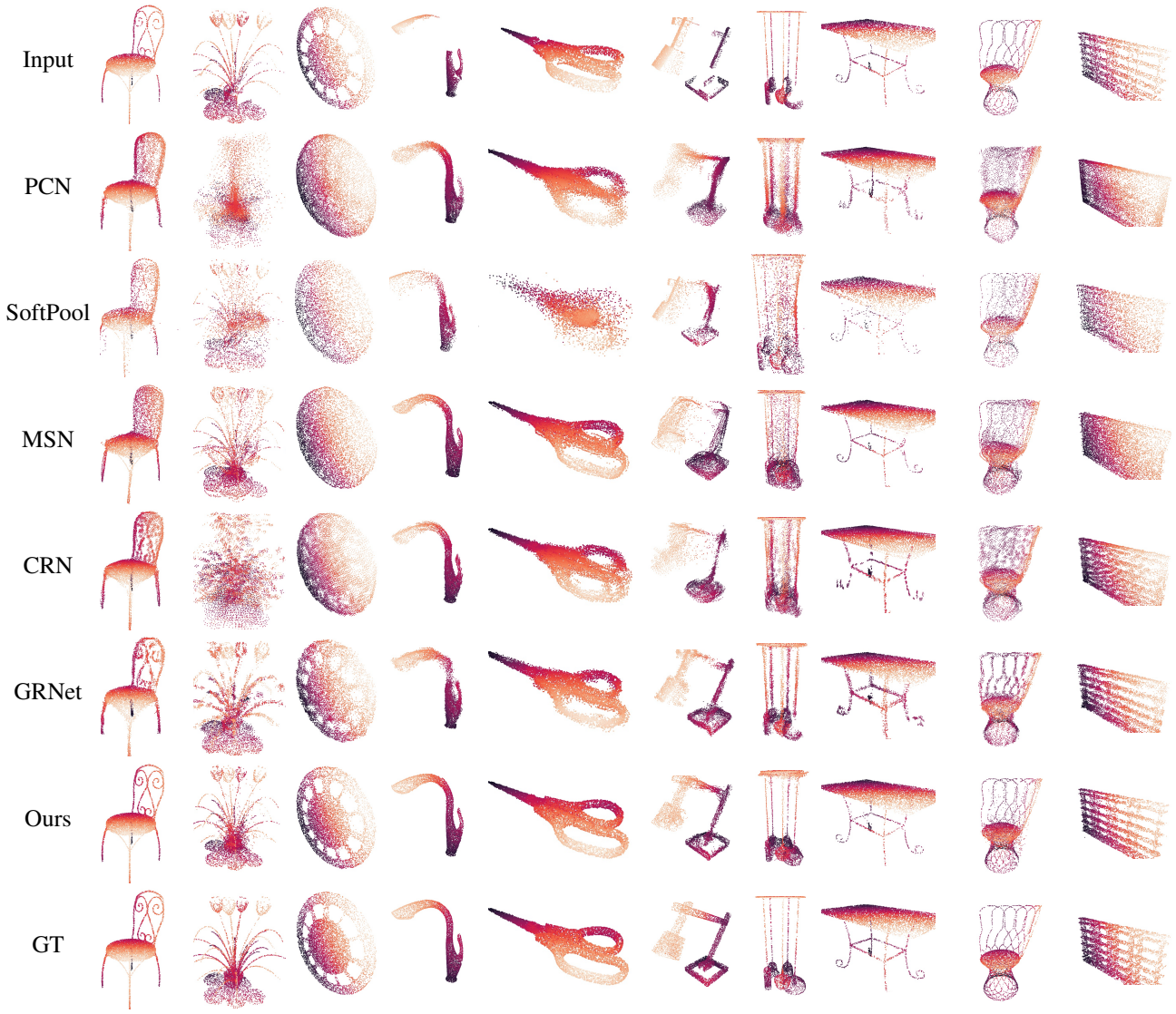


Figure 4: Comparisons of different methods on point cloud completion. Note that 8,192 points are exported from each method for comparison, except SoftPoolNet (4,096 points) due to its network specification.

them under the resolution of 8,192. In these methods, **PCN** completes the partial point cloud by using a stacked version of PointNet layers [15] to construct an auto-encoder. **CRN** combines local details of partial input with the global shape feature to synthesize detailed object shapes with a coarse-to-fine strategy. Similarly, **MSN** also recovers shapes from coarse-to-fine and involves a joint loss function to ensure even point distribution. **GRNet** is the recent approach that applies 3D convolutions to process shapes on volumetric grids. We report the comparisons using EMD [10] and CD scores [4] in Table 1 and Table 2 respectively.

For the other methods that do not support upsampling (including PF-Net [8], P2P-Net[30], SoftPoolNet[25]), we downsample the output of all the methods to the resolution of 2,048 to enable a fair comparison. **SoftPoolNet** employs a similar encoder as PCN [31] but aggregates soft pooling layers as the activation function instead of max-pooling. **PF-Net** designs a multi-resolution pyramid decoder to recover the missing geometries on different scales. **P2P-Net** generates compact completion results by learning the bidirectional deformation between the input partial point cloud and the complete point cloud, but it struggles to recover the detailed structure, especially on invisible areas. The quantitative scores on EMD and CD are respectively listed in Table 3 and Table 4.

As shown in Table 1-4, our method outperforms existing methods on both EMD and CD scores. Our method uses EMD as a loss function and archives the lowest EMD score

methods	faucet	cabinet	table	chair	vase	lamp	average
PCN	16.49	9.34	11.34	10.94	12.43	16.10	12.77
PCN+Ray	13.63	8.25	10.79	9.74	11.10	14.30	11.30
CRN	13.43	9.85	7.93	8.67	12.49	11.38	10.63
GRNet	10.36	7.75	7.50	7.74	11.21	10.74	9.22
MSN	7.71	6.70	6.52	6.57	6.89	7.55	6.99
Ours	**6.31**	**6.14**	**5.33**	**5.12**	**5.93**	**6.76**	**5.93**

Table 1: Evaluation on EMD ($\times 10^2$) with Res.=8,192

methods	faucet	cabinet	table	chair	vase	lamp	average
PCN	4.17	4.67	3.82	4.01	6.31	3.73	4.45
PCN+Ray	2.80	4.55	3.57	3.81	5.80	3.12	3.94
CRN	3.67	**4.49**	**3.44**	3.81	5.49	3.19	4.01
GRNet	3.28	4.66	3.73	3.94	5.53	3.52	4.11
MSN	4.02	5.75	4.61	4.81	5.71	4.34	4.87
Ours	**2.62**	4.72	3.76	**3.62**	**4.54**	**3.02**	**3.71**

Table 2: Evaluation on CD ($\times 10^2$) with Res.=8,192

methods	faucet	cabinet	table	chair	vase	lamp	average
PCN	16.81	10.47	12.22	11.81	13.25	16.67	13.54
PCN+Ray	16.13	10.18	11.68	10.61	11.13	14.90	12.44
PF-Net	16.11	10.04	9.97	10.61	11.50	14.07	12.05
P2P-Net	16.09	11.64	10.73	12.29	16.36	13.52	13.44
SoftPoolNet	15.03	14.30	11.28	14.05	17.63	15.89	14.70
CRN	14.00	11.00	9.09	9.70	13.32	12.09	11.53
GRNet	11.30	9.16	8.61	8.82	12.27	11.28	10.24
MSN	8.52	8.19	7.82	7.82	8.36	8.51	8.20
Ours	**6.89**	**7.48**	**6.63**	**6.63**	**7.16**	**7.48**	**7.05**

Table 3: Evaluation on EMD ($\times 10^2$) with Res.=2,048

methods	faucet	cabinet	table	chair	vase	lamp	average
PCN	5.62	7.28	5.95	6.14	8.71	5.15	6.48
PCN+Ray	4.35	7.14	5.19	5.98	7.19	4.61	5.74
PF-Net	8.96	8.15	6.94	7.48	10.10	7.56	8.20
P2P-Net	4.47	7.21	**5.49**	5.92	7.62	4.41	5.85
SoftPoolNet	5.54	7.85	6.41	6.59	8.27	5.56	6.70
CRN	5.14	7.13	5.59	5.94	7.96	4.63	6.06
GRNet	4.72	7.21	5.77	6.00	7.90	4.92	6.08
MSN	5.25	8.06	6.50	6.70	7.92	5.66	6.68
Ours	**3.90**	**7.01**	5.65	**5.61**	**6.68**	**4.26**	**5.51**

Table 4: Evaluation on CD ($\times 10^2$) with Res.=2,048

in all object categories. Besides, we can observe some categories present large scores among all methods, which indicates their inherent structure complexity such as *faucet*, *vase*, *chair*, and *lamp*. While our method shows superiority especially in those categories on both EMD and CD scores. The qualitative results in Figure 4 demonstrate that PointNet-based methods like PCN, SoftPool, and MSN fail to reconstruct subtle structures and tend to generate blurred details. This is due to the limitation of PointNet architecture. GRNet uses volumetric convolution and therefore can encode the emptiness semantic implicitly, result in better details. However, GRNet is still limited by high computational cost. With the help of ray features encoded in points, our method can encode and reconstruct complex structures precisely and efficiently.

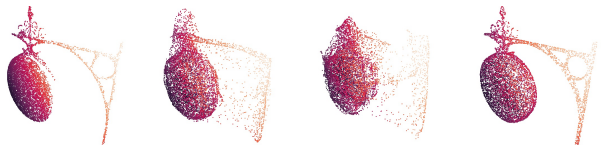
SK-PCN [13] predicts detailed structure by learning the skeletal points first. It decouples the shape completion into structure estimation and surface reconstruction. Limited by the availability of skeleton points for training, we only evaluate SK-PCN and ours on two categories: chair and table, where each of them has 818 and 991 models, respectively. The results are listed in Table 5.

Category	SK-PCN	Ours
Chair	8.46 / 4.75	4.01 / 3.09
Table	8.98 / 4.37	5.53 / 3.94
Average	8.72 / 4.56	4.77 / 3.51

Table 5: Comparison with SK-PCN (EMD / CD $\times 10^2$)

4.2. Effectiveness of Emptiness Encoding

To verify the effectiveness of embedding emptiness information in feature encoding, we compare the coarse output from MSN [10] (vanilla MSN) with the one embedded with ray features as in Section 3.3 (vanilla MSN+ray). The output resolution of the two methods is set to 8,192 points. We report the quantitative results in Table 6 on both EMD and CD scores. Besides, we also augment PCN [31] into PCN+ray. The results are listed in Table 1-4. Figure 5 shows a qualitative comparison. Both the qualitative and quantitative results suggest that, encoding emptiness feature improves the representation ability of the encoder, which further boosts the performance in shape decoding even using a coarse decoder (from a global feature). The reason could be that the emptiness feature indicates a clear shape contour. There could be fewer output points within empty areas from MSN+ray compared with using vanilla MSN.



(a) Input (b) v. MSN+ray (c) v. MSN (d) GT

Figure 5: Coarse results comparison with vanilla MSN+ray.

4.3. Encoding Emptiness with 2D Convolution

Apart from using 3D rays to encode emptiness, we also adopt 2D CNNs to perceive the empty regions on the depth map. In our experiments, the depth map in Figure 2 is concatenated with the emptiness $mask$ in Equation 1 to construct a 4-channel image I. Then we build a ME-PCN-2DConv network by replacing MLPs in the global encoder

Category	vanilla MSN	vanilla MSN+Ray
Faucet	8.61 / 5.03	6.59 / 3.63
Cabinet	7.17 / 6.07	6.13 / 5.24
Table	8.59 / 5.75	6.10 / 4.66
Chair	7.51 / 5.59	5.84 / 4.50
Vase	8.67 / 6.75	6.63 / 5.62
Lamp	8.59 / 5.42	7.84 / 4.38
Average	8.19 / 5.77	6.52 / 4.67

Table 6: MSN + emptiness encoding (EMD / CD $\times 10^2$)

Category	ME-PCN-2DConv	Ours
Faucet	7.46 / 3.31	6.31 / 2.62
Cabinet	6.26 / 5.00	6.14 / 4.72
Table	5.70 / 4.14	5.33 / 3.76
Chair	5.91 / 4.13	5.12 / 3.62
Vase	6.97 / 5.35	5.93 / 4.54
Lamp	6.74 / 3.54	6.76 / 3.02
Average	6.51 / 4.25	5.93 / 3.71

Table 7: 2D emptiness encoding (EMD / CD $\times 10^2$)

with three 2D convolution layers + Max-pooling, which outputs the global feature with the same dimension (2048-D) for the following shape decoding.

(a) Input (b) Ours (c) w. 2DConv (d) GT

Figure 6: Using 2D convolutions for emptiness encoding.

For coarse-to-fine decoding, we project predicted coarse points back to a 2D image plane. For each coarse point, we obtain its point feature by querying the 2D feature map learned by another 2D CNN from image I. The 2D CNN is constructed in a similar way as above, and obtains a 64-channel feature map from I. We compare ME-PCN-2DConv with our ME-PCN. The qualitative and quantitative results are listed in Figure 6 and Table 7. They show that ME-PCN presents better results. Besides, MEM-PCN-2DConv consumes more net parameters and costs 2-5 times of GPU memory led by the 2D convolutions, which demonstrates the efficiency of representing emptiness using rays.

4.4. Robustness of Emptiness Encoding

The emptiness masks in our method are generated from depth maps. However, in the real world, such a mask can be inaccurate considering the noises in depth scans. To verify our robustness to noisy masks, we simulate the masks from real-world depth/RGB data, and add strong Gaussian noise (see Figure 7) to the boundaries of masks. We fine-tune our model on chair and table categories using the noisy mask for 2000 iterations. Test results show that the EMD score on chair / table increase from 5.12 / 5.33 to 5.26 / 5.56 respectively. CD score on chair/table increase from 3.62 / 3.76 to 3.67 / 3.92 respectively. The performance-degradation is less than 3%, which verifies the robustness of our method.

(a) Before (b) After

Figure 7: Adding noise to simulate the mask from real world data: a) mask of chair back without noise; b) mask with strong noise on boundaries;

5. Conclusion

We provide a novel feature encoding modality for point completion, namely ME-PCN. It leverages the 3D emptiness in shape space to make neural networks sensitive to shape boundaries. In our method, we complete surface points by learning from both the shape occupancy and emptiness. A ray-based emptiness encoding strategy is proposed to perceive the emptiness clues on shape boundaries. It enables our method to recover enriched surface details while keeping consistent local topology. Verified by the ablation studies, our emptiness encoding is effective, moreover, robust and efficient. Extensive experiments demonstrate that our method achieves much better shape completion quality and largely outperforms the state-of-the-art on EMD and CD metrics. Though there is still some gap between our predictions and the GT, e.g., it may fail to capture small and complex topology, like decorations on the table stand, our method works well in general cases. We hope our work can serve as a universal improvement strategy for point completion and draw attention to the information in the 'emptiness', even in the larger community.

Acknowledgments This work was sponsored by Hong Kong Research Grants Council under General Research Funds (HKU17206218), and partially supported by NSFC-61902334 and Shenzhen General project JCYJ20190814112007258. It was also supported by CCF-Tencent AI Lab RhinoBird Fund.

References

[1] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[3] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. 2

[4] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2, 6

[5] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 2

[6] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 2

[7] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 85–93, Oct 2017. 2

[8] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 2, 5, 6

[9] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 2

[10] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11596–11603, Apr. 2020. 1, 2, 4, 5, 6, 7

[11] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[12] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 2

[13] Yinyu Nie, Yiqun Lin, Xiaoguang Han, Shihui Guo, Jian Chang, Shuguang Cui, and Jian.J Zhang. Skeleton-bridged point completion: From global inference to local adjustment. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16119–16130. Curran Associates, Inc., 2020. 2, 5, 7

[14] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9964–9973, 2019. 2

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 6

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1

[17] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 2

[18] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016. 2

[19] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 2

[20] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2

[21] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 1

[22] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 2

[23] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: a patch-based deep representation of 3d shapes. In *SIGGRAPH Asia 2018 Technical Papers*, page 217. ACM, 2018. 2

[24] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 790–799, June 2020. 2, 4, 5

[25] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpoolnet: Shape descriptor for point cloud completion and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 70–85, August 2020. 5, 6

[26] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1939–1948, 2020. 2

[27] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 5

[28] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4619–4628, June 2021. 2, 3

[29] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020. 2, 5

[30] Kangxue Yin, Hui Huang, Daniel Cohen-Or, and Hao Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 5, 6

[31] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1, 2, 4, 5, 6, 7