PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths

Xin Wen[1], Peng Xiang[1], Zhizhong Han[2], Yan-Pei Cao[3], Pengfei Wan[3], Wen Zheng[3], Yu-Shen Liu[1]*

[1]School of Software, BNRist, Tsinghua University, Beijing, China
[2]Department of Computer Science, Wayne State University, USA
[3] Y-tech, Kuaishou Technology, Beijing, China

{x-wen16,xp20}@mails.tsinghua.edu.cn h312h@wayne.edu caoyanpei@gmail.com

{wanpengfei,zhengwen}@kuaishou.com liuyushen@tsinghua.edu.cn

Abstract

The task of point cloud completion aims to predict the missing part for an incomplete 3D shape. A widely used strategy is to generate a complete point cloud from the incomplete one. However, the unordered nature of point clouds will degrade the generation of high-quality 3D shapes, as the detailed topology and structure of discrete points are hard to be captured by the generative process only using a latent code. In this paper, we address the above problem by reconsidering the completion task from a new perspective, where we formulate the prediction as a point cloud deformation process. Specifically, we design a novel neural network, named PMP-Net, to mimic the behavior of an earth mover. It moves move each point of the incomplete input to complete the point cloud, where the total distance of point moving paths (PMP) should be shortest. Therefore, PMP-Net predicts a unique point moving path for each point according to the constraint of total point moving distances. As a result, the network learns a strict and unique correspondence on point-level, and thus improves the quality of the predicted complete shape. We conduct comprehensive experiments on Completion3D and PCN datasets, which demonstrate our advantages over the state-of-the-art point cloud completion methods. Code will be available at https://github.com/diviswen/PMP-Net.

1. Introduction

As one of the widely used 3D shape representations, point cloud can be easily obtained through depth cameras or other 3D scanning devices. Due to the limitations of view-angles or occlusions of 3D scanning devices, the raw point clouds are usually sparse and incomplete [41]. Therefore, a
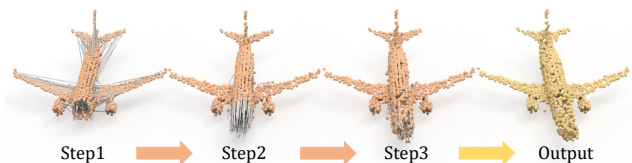
Figure 1. Illustration of completion by multi-step point cloud deformation. The point moving paths are denoted by grey lines. At each step, the source point cloud is colored by orange and the target point cloud is colored by yellow, respectively.

shape completion/consolidation process is usually required to generate the missing regions of 3D shape for the downstream 3D computer vision applications like classification [12, 6, 39, 10, 22, 21], segmentation [40, 17] and other visual analysis [47].

In this paper, we focus on the completion task for 3D objects represented by point clouds, where the missing parts are caused by self-occlusion due to the view angle of scanner. Most of the previous methods formulate point cloud completion as a point cloud generation problem [2, 41, 48, 34], where an encoder-decoder framework is usually adopted to extract a latent code from the input incomplete point cloud, and decode the extracted latent code into a complete point cloud. Benefiting from the deep learning based point cloud learning methods, the point cloud completion methods along this line have made huge progress in the last few years [41, 34]. However, the generation of point clouds remains a difficult task using deep neural network, because the unordered nature of point clouds makes the generative model difficult to capture the detailed topology or structure among discrete points [34]. Therefore, the completion quality of point clouds based on generative models is still unsatisfying.

To address this issue, in this paper, we propose a novel neural network, named PMP-Net, to formulate the task of point cloud completion from a new perspective. Different from the generative model that directly predicts the coordinations of all points in 3D space, the PMP-Net learns to
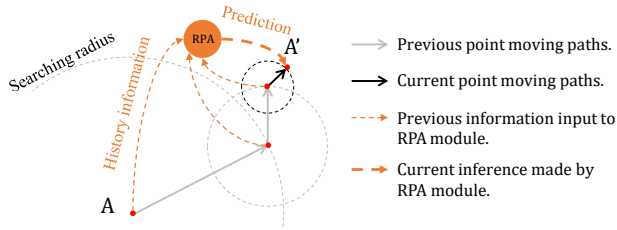
Figure 2. Illustration of path searching with multiple steps under the coarse-to-fine searching radius. The PMP-Net moves point A to point A' by three steps, with each step reducing its searching radius, and looking back to consider the moving history in order to decide the next place to move.

move the points from the source 3D shape to the target one. Through the point moving process, the PMP-Net establishes the point-level correspondences between the source point cloud and the target, which captures the detailed topology and structure relationships between the two point clouds. On the other hand, there are various solutions to move points from source to target, which will confuse the network during training. Therefore, in order to encourage the network to learn a unique arrangement of point moving path, we take the inspiration from the Earth Mover's Distance (EMD) and propose to regularize a Point-Moving-Path Network (PMP-Net) under the constraint of the total point moving distances (PMDs), which guarantees the uniqueness of path arrangement between the source point cloud and the target one.

Moreover, in order to predict the point moving path more accurately, we propose a multi-step path searching strategy to continuously refine the point moving path under multi-scaled searching radius. Specifically, as shown in Figure 2, the path searching is repeated for multiple steps in a coarse-to-fine manner. Each step will take the previously predicted paths into consideration, and then plan its next move path according to the previous paths. To record and aggregate the history information of point moving path, we take the inspiration from Gated Recurrent Unit (GRU) to propose a novel Recurrent Path Aggregation (RPA) module. It can memorize and aggregate the route sequence for each point, and combine the previous information with the current location of point to predict the direction and the length for the next move. By reducing the searching radius step-by-step, PMP-Net can consistently refine a more and more accurate path for each point to move from its original position on the incomplete point cloud to the target position on the complete point cloud. In all, the main contribution of our work can be summarized as follows.

- We propose a novel network for point cloud completion task, named PMP-Net, to move each point from the incomplete shape to the complete one to achieve a high quality point cloud completion. Compared with previous generative completion methods, PMP-Net has the ability to learn more detailed topology and structure relationships between incomplete shapes and complete ones, by learning the point-level correspondence through point moving path prediction.

- We propose to learn a unique point moving path arrangement between input and output point clouds, by regularizing the network using the constraint of Earth Mover's Distance. As a result, the network will not be confused by multiple solutions of moving points, and finally predicts a meaningful point-wise correspondence between the source and target point clouds.

- We propose to search point moving path with multiple steps in a coarse-to-fine manner. Each step will decide the next move based on the aggregated information from the previous paths and its current location, by using the proposed Recurrent Path Aggregation (RPA) module.

2. Related Work

The deep learning technology in 3D reconstruction [8, 5, 9, 18, 5] and representation learning [23, 11, 24, 7] have boosted the research of 3D shape completion, which can be roughly divided into two categories. (1) Traditional 3D shape completion methods [33, 1, 35, 15] usually formulate hand-crafted features such as surface smoothness or symmetry axes to infer the missing regions, while some other methods [30, 19, 25, 31] consider the aid of large-scale complete 3D shape datasets, and perform searching to find the similar patches to fill the incomplete regions of 3D shapes. (2) Deep learning based methods [50, 29, 16, 14], on the other hand, exploit the powerful representation learning ability to extract geometric features from the incomplete input shapes, and directly infer the complete shape according to the extracted features. Those learnable methods do not require the predefined hand-crafted features in contrast with traditional completion methods, and can better utilize the abundant shape information lying in the large-scale completion datasets. The proposed PMP-Net also belongs to the deep learning based method, where the methods along this line can be further categorized and detailed as below.

Volumetric aided shape completion. The representation learning ability of convolutional neural network (CNN) has been widely used in 2D computer vision research, and the studies concerning application of 2D image inpainting have been continuously surging in recent years. A intuitive idea for 3D shape completion can be directly borrowed from the success of 2D CNN in image inpainting research [46, 44, 20], extending it into 3D space. Recently, several volumetric aided shape completion methods, which are based on 3D CNN structure, have been developed. Note that we use the term *"volumetric aided"* to describe this kind of methods, because the 3D voxel is usually not the final

output of the network. Instead, the predicted voxel will be further refined and converted into other representations like mesh [2] or point cloud [42], in order to produce more detailed 3D shapes. Therefore, the voxel is more like an intermediate aid to help the completion network infer the complete shape. Notable works along this line like 3D-EPN [2] and GRNet [42] have been proposed to reconstruct the complete 3D voxel in a coarse-to-fine manner. They first predict a coarse complete shape using 3D CNN under an encoder-decoder framework, and then refine the output using similar patches selected from a complete shape dataset [2] or by further reconstructing the detailed point cloud according to the output voxel [42]. Also, there are some studies that consider purely volumetric data for shape completion task. For example, Han et. al [4] proposed to directly generate the high-resolution 3D volumetric shape, by simultaneously inferring global structure and local geometries to predict the detailed complete shape. Stutz et. al [32] proposed a variational auto-encoder based method to complete the 3D voxel under weak supervision. Despite the fascinating ability of 3D CNN for feature learning, the computational cost which is cubic to the resolution of input voxel data makes it difficult to process fine-grained shapes [41].

Point cloud based shape completion. There is a growing attention on the task of point cloud based shape completion [38, 34, 41, 13] in recent years. Since point cloud is a direct output form of many 3D scanning devices, and the storage and process of point clouds require much less computational cost than volumetric data, many recent studies consider to perform direct completion on 3D point clouds. Enlighten from the improvement of point cloud representation learning [27, 26], previous methods like TopNet [34], PCN [48] and SA-Net [41] formulate the solution as a generative model under an encoder-decoder framework. They adopted encoder like PointNet [26] or PointNet++ [27] to extract the global feature from the incomplete point cloud, and use a decoder to infer the complete point cloud according to the extracted features. Compare to PCN [48], TopNet [34] improved the structure of decoder in order to implicitly model and generate point cloud in a rooted tree architecture [34]. SA-Net [41] took one step further to preserve and convey the detailed geometric information of incomplete shape into the generation of complete shape through skip-attention mechanism. Other notable work like RL-GAN-Net [29] and Render4Completion [13] focused on the framework of adversarial learning to improve the reality and consistency of the generated complete shape. In all, most of the above methods are generative solution for point cloud completion task, and inevitably suffer from the unordered nature of point clouds, which makes it difficult to reconstruct the detailed typology or structure using a generative decoder. Therefore, in order to avoid the problem of predicting unordered data, PMP-Net uses a different way to reconstruct the complete point cloud, which learns to move all points from the initial input instead of directly generating the final point cloud from a latent code. The idea of PMP-Net is also related to the research of 3D shape deformation [45], which mainly considered one-step deformation. However, the deformation between the incomplete and complete shapes is more challenging, which requires the inference of totally unknown geometries in missing regions without any other prior information. In contrast, we propose multi-step searching to encourage PMP-Net to infer more detailed geometric information for missing region, along with point moving distance regularization to guarantee the efficiency of multi-step inference.

3. Architecture of PMP-Net

3.1. Point Displacement Prediction

Multi-step framework. An overview of the proposed PMP-Net is shown in Figure 3(a). Given an input point cloud $P = \{\boldsymbol{p}_i\}$ and a target point cloud $P' = \{\boldsymbol{p'}_j\}$. The objective of PMP-Net is to predict a displacement vector set $\Delta P = \{\Delta \boldsymbol{p}_i\}$, which can move each point from P into the position of P' such that $\{(\boldsymbol{p}_i + \Delta \boldsymbol{p}_i)\} = \{\boldsymbol{p'}_j\}$. PMP-Net moves each point $\boldsymbol{p}_i$ for $K = 3$ steps in total. The displacement vector for step k is denoted by $\Delta \boldsymbol{p}_i^k$, so $\Delta \boldsymbol{p}_i = \sum_{k=1}^{3} \Delta \boldsymbol{p}_i^k$. For step k, the network takes the deformed point cloud $\{\boldsymbol{p}_i^{k-1}\} = \{\boldsymbol{p}_i + \sum_{j=1}^{k-1} \Delta \boldsymbol{p}_i^j\}$ from the last step $k-1$ as input, and calculates the new displacement vector according to the input point cloud. Therefore, the predicted shape will be consistently refined step-by-step, which finally produces a complete shape with high quality.

Displacement vector prediction. At step k, in order to predict the displacement vector $\Delta \boldsymbol{p}_i^k$ for each point, we first extract per-point features from the point cloud. This is achieved by first adopting the basic framework of Point-Net++ [27] to extract the global feature of input 3D shape, and then using the feature propagation module to propagate the global feature to each point in 3D space, and finally producing per-point feature $\boldsymbol{h}_i^{k,l}$ for point $\boldsymbol{p}_i^k$. Since our experimental implementation applies three levels of feature propagation to hierarchically produce per-point feature (see Figure 3(b)), we use superscript k to denote the step and the subscript l to denote the level in $\boldsymbol{h}_i^{k,l}$. The per-point feature $\boldsymbol{h}_i^{k,l}$ is then concatenated with a random noise vector $\hat{\boldsymbol{x}}$, which according to [45] can give point tiny disturbances and force it to leave its original place. Then, the final point feature $\boldsymbol{h}_i^{k,3}$ at step k and level 3 is fed into a multi-layer perceptron (MLP) followed by a hyper-tangent activation (tanh), to produce a 3-dimensional vector as the displacement vector $\Delta \boldsymbol{p}_i^k$ for point $\boldsymbol{p}_i^k$ as

$$\Delta \boldsymbol{p}_i^k = \tanh(\text{MLP}([\boldsymbol{h}_i^{k,3} : \hat{\boldsymbol{x}}])), \tag{1}$$

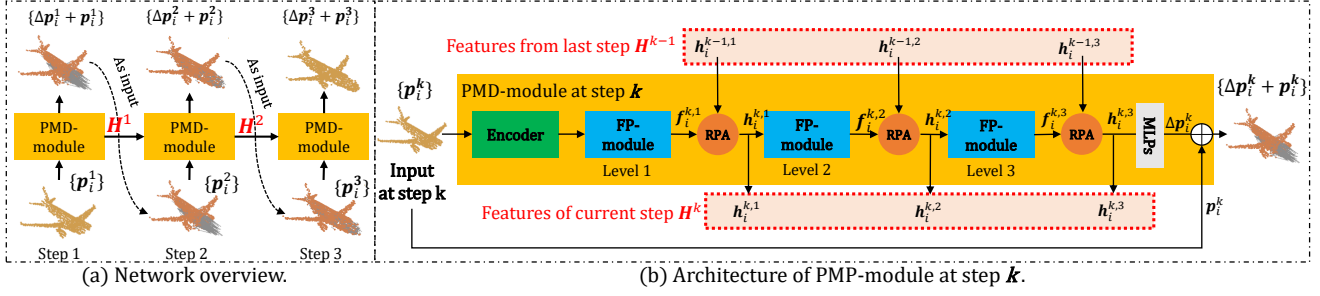where ":" denotes the concatenation operation.

Figure 3. Illustration of path searching with multiple steps under the coarse-to-fine searching radius. The PMP-Net moves point A to point A' by three steps, with each step reducing its searching radius, and looking back to consider the moving history in order to decide the next place to move.
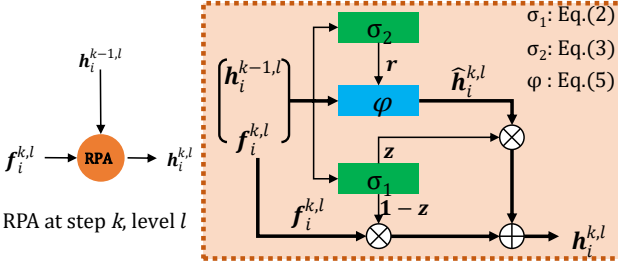

Figure 4. Detailed structure of RPA module at step k, level l.

Recurrent information flow between steps. The information of previous moves is crucial for network to decide the current move, because the previous paths can be used to infer the location of the final destination for a single point. Moreover, such information can guide the network to find the direction and distance of next move, and prevent it from changing destination during multiple steps of point moving path searching. In order to achieve this target, we propose to use a special RPA unit between each step and each level of feature propagation module, which is used to memorize the information of previous path and to infer the next position of each point. As shown in Figure 3(b), the RPA module in (step k, level l) takes the output $\boldsymbol{f}_i^{k,l-1}$ from the last level i-1 as input, and combines it with the feature $\boldsymbol{h}_i^{k-1,l}$ from the previous step $k-1$ at the same level l to produce the feature of current level $\boldsymbol{h}_i^{k,l}$, denoted as

$$\boldsymbol{h}_i^{k,l} = \mathrm{RPA}(\boldsymbol{f}_i^{k,l-1}, \boldsymbol{h}_i^{k-1,l}). \tag{2}$$

The detailed structure of RPA module is described below.

3.2. Recurrent Path Aggregation

The previous paths of point moving can be regarded as the sequential data, where the information of each move should be selectively memorized or forgotten during the process. Following this idea, we take the inspiration from the recurrent neural network, where we mimic the behavior of gated recurrent unit (GRU) to calculate an update gate $\boldsymbol{z}$ and reset gate $\boldsymbol{r}$ to encode and forget information, which is according to the point feature $\boldsymbol{h}_i^{k-1,l}$ from the last step $k-1$ and the point feature $\boldsymbol{f}_i^{k,l-1}$ of current step k. The

calculation of two gates can be formulated as

$$\boldsymbol{z} = \sigma(W_z[\boldsymbol{f}_i^{k,l-1} : \boldsymbol{h}_i^{k-1,l}] + \boldsymbol{b}_z), \tag{3}$$

$$\boldsymbol{r} = \sigma(W_r[\boldsymbol{f}_i^{k,l-1} : \boldsymbol{h}_i^{k-1,l}] + \boldsymbol{b}_r), \tag{4}$$

where W_z, W_r are weight matrix and $\boldsymbol{b}_z, \boldsymbol{b}_z$ are biases. σ is the *sigmoid* activation function, which predicts a value between 0 and 1 to indicate the ratio of information that allowed to pass the gate. ":" denotes the concatenation of two features.

Different from the standard GRU, which emphasizes more importance on the preservation of previous information when calculating the output feature $\boldsymbol{h}_i^{k,l}$ at current step, in RPA, we address more importance on the preservation of current input information, and propose to calculate the output feature $\boldsymbol{h}_i^{k,l}$ as

$$\boldsymbol{h}_i^{k,l} = \boldsymbol{z} \odot \hat{\boldsymbol{h}}_i^{k,l} + (1 - \boldsymbol{z}) \odot \boldsymbol{f}_i^{k,l-1}, \tag{5}$$

where $\hat{\boldsymbol{h}}_i^{k,l}$ is the intermediate feature of current step. It contains the preserved information from the past, which is calculated according to the current input feature. The formulation of $\hat{\boldsymbol{h}}_i^{k,l}$ is given as

$$\hat{\boldsymbol{h}}_i^{k,l} = \varphi(W_h[\boldsymbol{r} \odot \boldsymbol{h}_i^{k-1,l} : \boldsymbol{f}_i^{k,l-1}] + \boldsymbol{b}_h), \tag{6}$$

where φ is *relu* activation in our implementation.

The reason of fusing $\hat{\boldsymbol{h}}_i^{k,l}$ with $\boldsymbol{f}_i^{k,l-1}$ instead of $\boldsymbol{h}_i^{k-1,l}$ is that, compared with standard unit in RNN unit, the current location of point should have greater influence to the decision of next move. Especially, when RPA module needs to ignore the previous information which is not important in the current decision making, Eq.(5) can easily allow RPA model to forget all history by simply pressing the update gate $\boldsymbol{z}$ to a zero-vector, and thus enables the RPA module fully focus on the information of current input $\boldsymbol{f}_i^{k,l-1}$.

3.3. Optimized Searching for Unique Paths

Minimizing moving distance. As shown in Figure 5, the unordered nature of point cloud allows multiple solutions to deform the input shape into the target one, and the direct

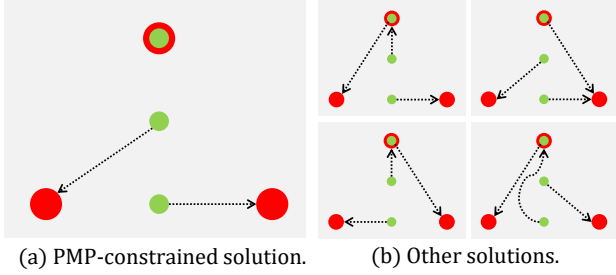(a) PMP-constrained solution.　　(b) Other solutions.

Figure 5. Illustration of multiple solutions when deforming input point cloud (green) into target point cloud (red). The PMD-constraint guarantees the uniqueness of point level correspondence (a) between input and target point cloud, and filter out various redundant solutions for moving points (b).

constraint (e.g. chamfer distance) on the deformed shape and its ground truth cannot guarantee the uniqueness of correspondence established between the input point set and the target point set. Otherwise, the network will be confused by the multiple solutions of point moving, which may lead to the failure of capturing detailed topology and structure relationships between incomplete shapes and complete ones. In order to establish a unique and meaningful point-wise correspondence between input point cloud and target point cloud, we take the inspiration from Earth Mover's Distance [28], and propose to train PMP-Net to learn the path arrangement ϕ between source and target point clouds under the constraint of total point moving path distance. Specifically, given the source point clouds $\hat{X} = \{\hat{\boldsymbol{x}}_i | i = 1, 2, 3, ..., N\}$ and the target point cloud $X = \{\boldsymbol{x}_i | i = 1, 2, 3, ..., N\}$, we follow EMD to learn an arrangement ϕ which meets the constraint below

$$\mathcal{L}_{\text{EMD}}(\hat{X}, X) = \min_{\phi:\hat{X} \to X} \frac{1}{\hat{X}} \sum_{\hat{\boldsymbol{x}} \in \hat{X}} \|\hat{\boldsymbol{x}} - \phi(\hat{\boldsymbol{x}})\|, \qquad (7)$$

In Eq.(7), ϕ is considered as a bijection that minimizes the average distance between corresponding points in $\hat{X}$ and X.

According to Eq.(7), bijection ϕ established by the network should achieve the minimum moving distance to move points from input shape to target shape. However, even if the correspondence between input and target point clouds is unique, there still exist various paths between source and target points, as shown in Figure 6. Therefore, in order to encourage the network to learn an optimal point moving path, we choose to minimize the point moving distance loss ($\mathcal{L}_{\text{PMD}}$), which is the sum of all displacement vector $\{\Delta\boldsymbol{p}_i^k\}$ output by all three steps in PMP-Net. The *Point Moving Distance* loss is formulated as

$$\mathcal{L}_{\text{PMD}} = \sum_k \sum_i \|\Delta\boldsymbol{p}_i^k\|_2. \qquad (8)$$

Eq.(8) is more strict than EMD constraint. It requires not only the overall displacements of all point achieve the shortest distance, but also limits the point moving paths in each
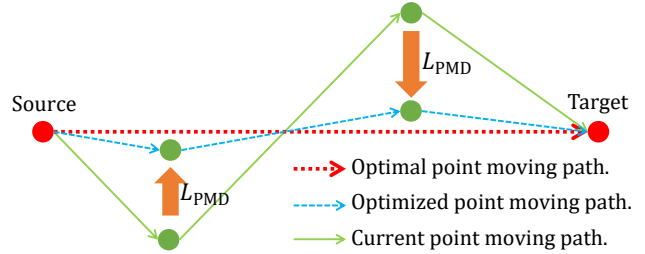

Figure 6. Illustration of the effectiveness of $\mathcal{L}_{PMD}$. By minimizing the point moving distance, the network is encouraged to learn more consistent paths from source to target, which will reduce redundant searching in each step and improve the efficiency.

step to be the shortest one. Therefore, in each step, the network will be encouraged to search new path following the previous direction, as shown in Figure 6, which will lead to less redundant moving decision and improve the searching efficiency.

Multi-scaled searching radius. PMP-Net searches the point moving path in a coarse-to-fine manner. For each step, PMP-Net reduces the maximum stride to move a point by the power of 10, which is, for step k, the displacement $\Delta\boldsymbol{p}_i^k$ calculated in Eq.(1) is limited to $10^{-k+1}\Delta\boldsymbol{p}_i^k$. This allows the network converges more quickly during training. And also, the reduced searching range will guarantee the network at next step not to overturn its decision made in the previous step, especially for the long range movements. Therefore, it can prevent the network from making redundant decision during path searching process.

3.4. Training Loss

The deformed shape is regularized by the complete ground truth point cloud through Chamfer distance (CD) and Earth Mover Distance (EMD). Following the same notations in Eq.(7), the Chamfer distance is defined as:

$$\mathcal{L}_{CD}(X, \hat{X}) = \sum_{\boldsymbol{x} \in X} \min_{\hat{\boldsymbol{x}} \in \hat{X}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\| + \sum_{\hat{\boldsymbol{x}} \in \hat{X}} \min_{\boldsymbol{x} \in X} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|. \quad (9)$$

The total loss for training is then given as

$$\mathcal{L} = \sum_k \mathcal{L}_{\text{CD}}(P^k, P') + \mathcal{L}_{\text{PMD}}, \qquad (10)$$

where P^k and P' denote the point cloud output by step k and the target complete point cloud, respectively. Note that finding the optimal ϕ is extremely computational expensive. In experiments, we follow the simplified algorithm in [48] to estimate an approximation of ϕ.
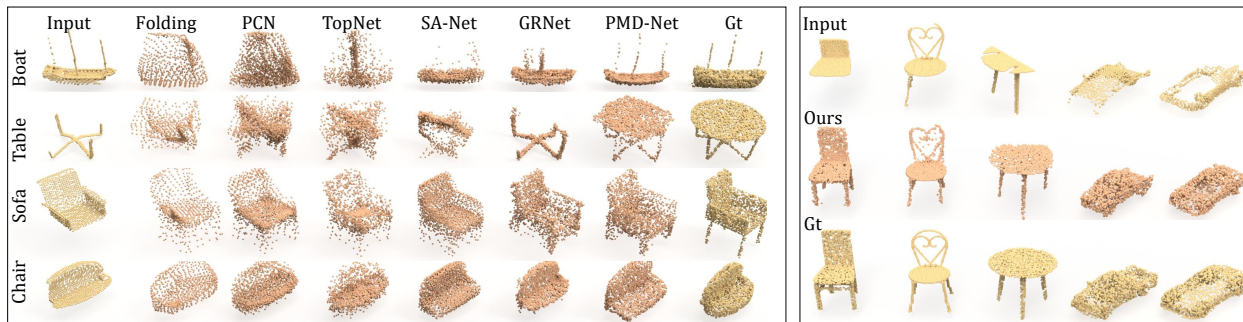
4. Experiments

4.1. Evaluation on Completion3D Dataset

Dataset. We evaluate our PMP-Net on the widely used benchmark of 3D point cloud completion, i.e. *Comple-*

Table 1. Point cloud completion on Completion3D dataset in terms of per-point L2 Chamfer distance $\times 10^4$ (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
FoldingNet [43]	19.07	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51
PCN [48]	18.22	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73
PointSetVoting [49]	18.18	6.88	21.18	15.78	22.54	18.78	28.39	19.96	11.16
AtlasNet [3]	17.77	10.36	23.40	13.40	24.16	20.24	20.82	17.52	11.62
SoftPoolNet [37]	16.15	5.81	24.53	11.35	23.63	18.54	20.34	16.89	7.14
TopNet [34]	14.25	7.32	18.77	12.88	19.82	14.60	16.29	14.89	8.82
SA-Net [41]	11.22	5.27	**14.45**	**7.78**	13.67	13.53	14.22	11.75	8.84
GRNet [42]	10.64	6.13	16.90	8.27	12.23	10.22	14.93	10.08	5.86
PMP-Net(Ours)	**9.23**	**3.99**	14.70	8.55	**10.21**	**9.27**	**12.43**	**8.51**	**5.77**

(a) Visualized completion comparison on ShapeNet.

(b) More completion results of PMD-Net on ShapeNet.

Figure 7. Visualization of point cloud completion results on Completion3D dataset. PMP-Net is compare with other methods in (a), and (b) shows more completion results of PMP-Net.

tion3D [34], which is a large-scaled 3D object dataset derived from the ShapeNet dataset. The partial 3D shapes are generated by back-projecting 2.5D depth images from partial views into 3D space.We follow the settings of training/validation/test splits in Completion3D for fair comparison with the other methods.

Evaluation metric. Following previous studies [34, 42, 41, 48], we use the per-point L1/L2 Chamfer distance (CD) as the evaluation metric. L1 version is the CD in Eq.(9) averaged by the point number, and L2 version is the CD replacing L1-norm in Eq.(9) by L2-norm.

Quantitative comparison. The comparison results[1] of PMP-Net with the other state-of-the-art point cloud completion methods is shown in Table 1, in which the PMP-Net achieves the best performance in terms of average chamfer distance across all categories. Compared with the second best completion method GRNet (in terms of average chamfer distance), PMP-Net achieves better results in 6 out of 8 categories, which proves the better generalization ability of PMP-Net across different shape categories. As we discussed in Sec.2, GRNet [42] is a voxel aided shape completion method, where the conversion between point clouds and 3D voxel along with applying 3D CNN on voxel data is a time consuming process. Other methods like SA-Net [41] in Table 1 are typical generative completion methods which are fully based on point clouds, and the nontrivial

improvement of PMP-Net over these methods proves the effectiveness of deformation based solution in point cloud completion task.

Qualitative comparison. In Figure 7(a), we visually compare PMP-Net with the other completion methods on Completion3D dataset, from which we can find that PMP-Net predicts much more accurate complete shapes on various input categories, while the other methods may output some failure cases for certain input shapes. For example, the input *table* in Figure 7(a) loses the entire desktop, and methods like GRNet and SA-Net almost cannot predict the whole desktop, while other methods like FoldingNet [43], PCN [48] and TopNet [34] intend to repair the desktop but fail to predict a good shape of it. Moreover, the advantage of deformation based PMP-Net over the generative methods can be well proved by the case of *boat* in Figure 7(a). Generative methods, especially like PCN and TopNet, successfully learn the overall structure of the input boat, but fail to reconstruct even the residual part of it. On the other hand, the deformation based PMP-Net can directly preserve the input shape by moving small amount of point to perform completion on certain areas, and keep the input shape unchanged. In Figure 7(b), we visualize some more completion results of PMP-Net.

Qualitative comparison on ScanNet chairs. To evaluate the generalization ability of PMP-Net on point cloud completion task, we pre-train PMP-Net on Completion3D dataset and evaluate its performance on the chair instances

[1]Results are cited from https://completion3d.stanford.edu/results

Table 2. Point cloud completion on PCN dataset in terms of per-point L1 Chamfer distance $\times 10^3$ (lower is better).

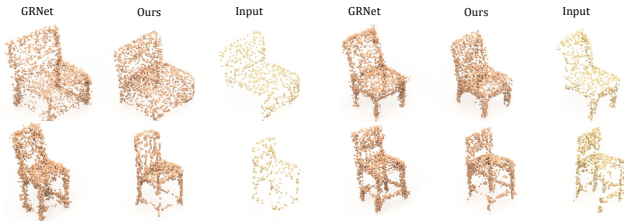| Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
FoldingNet [43]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet [34]	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet [3]	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
PCN [48]	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
GRNet [42]	8.83	6.45	10.37	9.45	**9.41**	7.96	**10.51**	8.44	8.04
CDN. [36]	**8.51**	**4.79**	**9.97**	**8.31**	9.49	8.94	10.69	**7.81**	8.05
PMP-Net(Ours)	8.66	5.50	11.10	9.62	9.47	**6.89**	10.74	8.77	**7.19**

Figure 8. Visual comparison of PMP-Net and GRNet on ScanNet chairs.

in ScanNet dataset without finetune, and compare with GR-Net (which is the second best method in Table 1, and also pre-trained on Completion3D). The visual comparison is shown in Figure 8. The PMP-Net completes shapes with less noise than GRNet, which benefits from its point moving practice. Because the differences in data distribution between Completion3D and ScanNet will inevitably confuse the network, and the point moving based PMP-Net can simply choose to leave those points in residual part of an object to stay at their own place to preserve a better shape, in contrast, generation based GRNet has to predict new points for both residual and missing part of an object.

4.2. Applying to Dense Point Cloud Completion

We show that PMP-Net learned on sparse point cloud can be directly applied to the dense point cloud completion. Specifically, we keep training PMP-Net on sparse shape with 2,048 points, and reveal its generalization ability by predicting dense complete shape with 16,384 points on PCN dataset [48]. PCN dataset is derived from ShapeNet dataset, in which each complete shape contains 16,384 points. The partial shapes have various point numbers, so we first down-sample shapes with more than 2,048 points to 2,048, and up-sample shapes with less than 2048 points to 2048 by randomly copying points. Since PMP-Net learns to move points instead of generating points, it requires the same number of points in incomplete point cloud and complete one. In order to predict complete shape of 16,384 points, we repeat 8 times of prediction for each shape during testing, with each time moving 2,048 points. Note that PMP-Net is still trained on sparse point clouds with 2,048 points, which are sampled from the dense point clouds of PCN dataset.

The comparison in Table 2 shows that PMP-Net yields a comparable performance to the state-of-the-art method

[36], and ranks second on PCN dataset. The result of Wang et al.[36] is cited from its original paper, while the results of other compared methods are all cited from [42]. Note that most generation based methods (like Wang et al. [36] and GRNet [42] in Table 2) specially designed a coarse-to-fine generation process in order to obtain better performance on dense point cloud completion. In contrast, our PMP-Net trained on 2,048 points can directly generate arbitrary number of dense points by simply repeating the point moving process, and still achieves comparable results to the counterpart methods.

4.3. Model Analysis

In this subsection, we analyze the influence of different parts in the PMP-Net. By default, we use the same network sittings in all experiments except for the analyzed part. All studies are typically conducted on the validation set of Completion3D dataset under four categories (i.e. plane, car, chair and table) for convenience.

Analysis of RPA module and PMP loss. We analyze the effectiveness of RPA module by replacing it with other units in PMP-Net. And for PMP loss, we analyze its effectiveness by removing PMP loss from the network. Specifically, we develop six different variations for comparison: (1) *NoPath* is the variation that removes the RPA module from the network; (2) *Add* is the variation that replaces RPA module with element-wise add layer in the network; (3) RNN, (4) LSTM and (5) GRU are variations that replace RPA module with different recurrent unit; (6) *w/o PMP* is the variation that removes PMP loss from the PMP-Net, where only Chamfer distance is used for training.

The shape completion results are shown in Table 3, from which we can find that baseline RPA module achieves the best performance. The worst results yielded by *Add* variation indicate that directly utilizing history information (paths in previous step) without processing will in return degenerate the network performance, compared to *NoPath* variation. The comparison between *RPA* baseline and *GRU* variation proves the effectiveness of our designation of RPA module, which can give more consideration to the information from current step than GRU unit, and help the network to make more precise decision for point moving.

By comparing baseline variation with *w/o PMP* varia-

tion, we can find that PMP loss significantly improves the performance of our network, which is in accordance with our opinion that point moving path should be regularized to better capture the detailed topology and structure of 3D shapes.

Table 3. Analysis of RPA and PMP loss (baseline marked by "*").

Unit.	avg.	plane	chair	car	table
NoPath	11.95	3.55	8.30	16.15	19.79
Add	12.23	**3.32**	8.10	16.47	21.05
RNN	12.12	3.55	8.14	16.19	20.58
LSTM	11.99	3.79	8.09	15.37	20.72
GRU	11.87	3.44	7.85	15.44	20.72
w/o PMP	13.66	4.26	8.21	16.69	25.19
baseline*	**11.58**	3.42	**7.87**	**15.88**	**19.15**

Effect of multi-step path searching. In Table 4, we analyze the effect of different steps for point cloud deformation. Specifically, we fix the ratio of searching radius between each step to 10, and evaluate the performance of PMP-Net under different step sittings. For example, the searching radius for step=4 is set to $\{1.0, 10^{-1}, 10^{-2}, 10^{-3}\}$, and the searching radius for step=2 is set to $\{1.0, 10^{-1}\}$, respectively. From Table 4, we can find that deforming point cloud by multiple steps effectively improves the completion performance, by comparing the results of step 1, 2 and 3. On the other hand, the comparison between step 3 and step 4 shows that the performance of multi-step path searching will reach its limitation, because too many steps may cause information redundancy in path searching.

Table 4. The effect of different steps (baseline marked by "*").

Steps.	avg.	plane	chair	car	table
1	12.26	3.71	8.27	15.59	21.48
2	11.90	3.47	7.95	15.66	20.53
3*	**11.58**	3.42	**7.87**	**15.88**	**19.15**
4	11.67	**3.39**	7.91	15.89	19.48

Analysis of searching radius. By default, we decrease the searching radius for each step by the ratio of 10. In Table 5, we analyze different sittings of searching radius and evaluate their influence to the performance of PMP-Net. We additionally test two different strategies to perform point moving path searching, i.e. the strategy without decreasing searching radius ([1.0, 1.0, 1.0] for each step), and the strategy with smaller decreasing ratio ([1.0, 0.5, 0.25]). The baseline result is the default sitting of PMP-Net ([1.0, 0.1, 0.01] for each step). Table 5 shows that PMP-Net achieves the worst performance at [1.0, 1.0, 1.0], which proves the effectiveness of the strategy to decrease searching radius. And when comparing strategy of [1.0, 0.5, 0.25] with [1.0, 0.1, 0.01], we can find that decreasing searching radius with larger ratio can improve the model performance, because larger ratio can better pre-

(a) Searching radius 1.0→0.1→0.01

(b) Searching radius 1.0→0.5→0.25

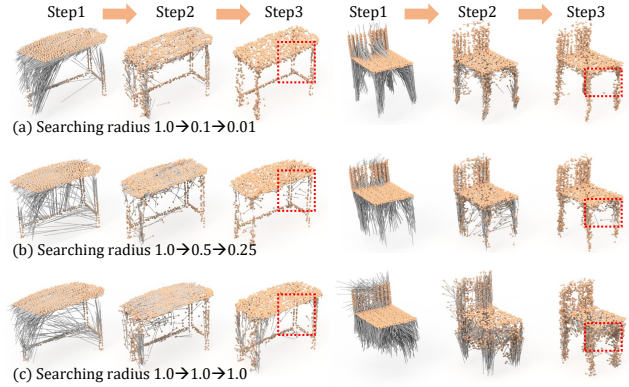(c) Searching radius 1.0→1.0→1.0

Figure 9. Illustration of deformation process in each step under different strategies of searching radius.

vent the network from overturning the decisions in previous steps. We also note that when the decreasing ratio goes to large, the PMP-Net will approximate the behavior of network with step=1 in Table 4, which can in return harm the performance of shape completion.

Table 5. The effect of searching radius (baseline marked by "*").

Radius.	Avg.	Plane	Chair	Car	Table
[1.0, 1.0, 1.0]	12.01	3.61	8.22	16.44	19.79
[1.0, 0.5, 0.25]	11.77	**3.36**	8.01	15.92	19.79
[1.0, 0.1, 0.01]*	**11.58**	3.42	**7.87**	**15.88**	**19.15**

Visual analysis of point moving path under different radius. In Figure 9, we visualize the searching process under different strategies of searching radius in Table 4. By analyzing the deformation output in step 1, we can find that PMP-Net with a coarse-to-fine searching strategy can learn to predict a better shape at early step, where the output of step 1 in Figure 9(a) is more complete and tidy than the ones in Figure 9(b) and Figure 9(c). Moreover, a better overall shape predicted in the early stage will enable the network focus on refining a better detailed structure of point cloud, which can be concluded from the comparison of step 3 in Figure 9, where the region highlighted by red rectangles in Figure 9(a) is much better than the other two subfigures.

5. Conclusions

In this paper, we propose a novel PMP-Net for point cloud completion by multi-step shape deformation. By moving points from source to target point cloud with multiple steps, PMP-Net can consistently refine the detailed structure and topology of the predicted shape, and establish the point-level shape correspondence between the incomplete and the complete shape. In experiments, we show the superioty of PMP-Net by comparing with other methods on the Completion3D benchmark and PCN dataset.

References

[1] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, Andrei Sharf, and Claudio Silva. State of the art in surface reconstruction from point clouds. In *Proceedings of the Conference of the European Association for Computer Graphics*, volume 1, pages 161–185, 2014. 2

[2] 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. 1, 3

[3] Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *CVPR 2018*, 2018. 6, 7

[4] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 85–93, 2017. 3

[5] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning (ICML)*, 2020. 2

[6] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *International Joint Conference on Artificial Intelligence*, 2019. 1

[7] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019. 2

[8] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. Reconstructing 3D shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 2020. 2

[9] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[10] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Junwei Han, Matthias Zwicker, and C.L. Philip Chen. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by RNN with attention. *IEEE Transactions on Image Processing*, 28(2):658–672, 2019. 1

[11] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-Angle Point Cloud-VAE: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10442–10451, 2019. 2

[12] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and CL Chen. 3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *International Joint Conference on Artificial Intelligence*, 2019. 1

[13] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. In *Proceedings of International Conference on Computer Vision*, 2019. 3

[14] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3D shape completion with multi-view consistent inference. In *AAAI*, 2020. 2

[15] Wei Hu, Zeqing Fu, and Zongming Guo. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Transactions on Image Processing*, 28(8):4087–4100, 2019. 2

[16] 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

[17] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020. 1

[18] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[19] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics*, 31(4):55, 2012. 2

[20] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2

[21] Xinhai Liu, Zhizhong Han, Fangzhou Hong, Yu-Shen Liu, and Matthias Zwicker. LRC-Net: Learning discriminative features on point clouds by encoding local region contexts. In *The 14th International Conference on Geometric Modeling and Processing*, 2020. 1

[22] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *33rd AAAI Conference on Artificial Intelligence*, 2019. 1

[23] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3D shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021. 2

[24] Xinhai Liu, Zhizhong Han, Xin Wen, Yu-Shen Liu, and Matthias Zwicker. L2G Auto-Encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 989–997. ACM, 2019. 2

[25] Andelo Martinovic and Luc Van Gool. Bayesian grammar learning for inverse procedural modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 201–208. IEEE, 2013. 2

[26] 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*, 2017. 3

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *In Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 3

[28] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The Earth Mover's Distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 5

[29] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. RL-GAN-Net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5898–5907, 2019. 2, 3

[30] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics*, 31(6):136, 2012. 2

[31] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics*, 31(6):180, 2012. 2

[32] 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. 3

[33] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015. 2

[34] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. TopNet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 1, 3, 6, 7

[35] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A field model for repairing 3D shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5684, 2016. 2

[36] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 7

[37] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. SoftPoolNet: Shape descriptor for point cloud completion and classification. In *ECCV*, 2020. 6

[38] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3

[39] Xin Wen, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu. Point2SpatialCapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. *IEEE Transactions on Image Processing*, 29:8855–8869, 2020. 1

[40] Xin Wen, Zhizhong Han, Geunhyuk Youk, and Yu-Shen Liu. CF-SIS: Semantic-Instance segmentation of 3D point clouds by context fusion with self-attention. In *ACM International Conference on Multimedia (ACM MM)*, 2020. 1

[41] 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. 1, 3, 6

[42] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. GRNet: Gridding residual network for dense point cloud completion. In *ECCV*, 2020. 3, 6, 7

[43] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 6, 7

[44] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017. 2

[45] 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. 3

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 2

[47] Jun Yuan, Changjian Chen, Weikai Yang, Mengchen Liu, Jiazhi Xia, and Shixia Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1), 2021. 1

[48] 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, 3, 5, 6, 7

[49] Junming Zhang, Weijia Chen, Yuping Wang, Ram Vasudevan, and Matthew Johnson-Roberson. Point set voting for partial point cloud analysis. *arXiv preprint arXiv:2007.04537*, 2020. 6

[50] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *ECCV*, 2020. 2