AnchorFormer: Point Cloud Completion from Discriminative Nodes

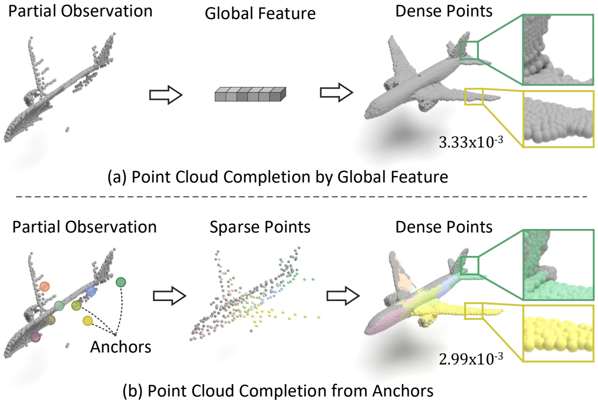
Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo and Tao Mei
University of Science and Technology of China, Hefei, China
University of Rochester, Rochester, NY USA
HiDream.ai Inc.
czk654@mail.ustc.edu.cn, {longfc.ustc, zhaofanqiu, tingyao.ustc}@gmail.com
zhwg@ustc.edu.cn, jluo@cs.rochester.edu, tmei@hidream.ai

Abstract

Point cloud completion aims to recover the completed 3D shape of an object from its partial observation. A common strategy is to encode the observed points to a global feature vector and then predict the complete points through a generative process on this vector. Nevertheless, the results may suffer from the high-quality shape generation problem due to the fact that a global feature vector cannot sufficiently characterize diverse patterns in one object. In this paper, we present a new shape completion architecture, namely AnchorFormer, that innovatively leverages pattern-aware discriminative nodes, i.e., anchors, to dynamically capture regional information of objects. Technically, AnchorFormer models the regional discrimination by learning a set of anchors based on the point features of the input partial observation. Such anchors are scattered to both observed and unobserved locations through estimating particular offsets, and form sparse points together with the down-sampled points of the input observation. To reconstruct the fine-grained object patterns, AnchorFormer further employs a modulation scheme to morph a canonical 2D grid at individual locations of the sparse points into a detailed 3D structure. Extensive experiments on the PCN, ShapeNet-55/34 and KITTI datasets quantitatively and qualitatively demonstrate the efficacy of AnchorFormer over the state-of-the-art point cloud completion approaches. Source code is available at https://github.com/chenzhik/AnchorFormer.

1. Introduction

As a 3D data description, point cloud can characterize various attributes of real-world objects. Although the point cloud data is readily acquired via laser scanners or depth cameras, factors like occlusion, transparency of surface, or the limit of sensor resolution, often cause geometric information loss and result in incomplete point cloud. As a result, it is an essential task of point cloud completion to improve the data quality for the downstream tasks, e.g., point cloud classification [21–23] and 3D object detection [20].



(a) Point Cloud Completion by Global Feature

(b) Point Cloud Completion from Anchors

Figure 1. An illustration of point cloud completion by leveraging (a) a global feature vector and (b) anchors in our AnchorFormer. We highlight the reconstruction results of two local patterns in the bounding boxes and visualize the details in the zoomed-in view. The L1 Chamfer Distance are also given.

Recent works [1, 34, 41, 46, 47] on point cloud completion usually formulate the task as a generation problem and mainly capitalize on an encoder-decoder architecture. The input partial points are encoded as a global feature vector which is further decoded to reconstruct the point cloud. Figure 1(a) conceptually depicts a typical process of point cloud completion through leveraging a global feature vector, which is generally measured by the pooling operation in the encoding phase to encapsulate the holistic shape information. The pooling operation inevitably leads to the loss of the fine-grained details and limits the capability of the global feature. It is thus difficult to decode from such degenerated global feature vector to reconstruct the diverse patterns of a 3D object, especially for completing some geometric details, e.g., the airplane tail in the green box in Figure 1(a). In contrast, we propose to rebuild object shape from a set of discriminative nodes, i.e., anchors, which indicate the local geometry of different patterns in an object, as shown in Figure 1(b). We derive the anchors from the input partial observation via self-attention. As such, the anchors could adequately infer the key patterns of the observed points. Moreover, some anchors can even be scattered into the unobserved locations to represent the missing parts and potentially capable of holistically reconstructing all patterns in the object. Taking the anchors and the down-sampled points of the input observation as the sparse points, we reform the fine-grained shape structure at the location of each sparse point to complete the point cloud of the object.

By exploiting the idea of shape reconstruction from pattern-aware discriminative nodes, we present a novel Anchor-based Transformer architecture namely AnchorFormer for point cloud completion. Given the input partial observation of a 3D object, AnchorFormer first down-samples the points and extracts the point features via an EdgeConv-based head [32]. Next, a transformer encoder takes the point features of down-sampled points as the inputs and is learnt to predict a set of coordinates, i.e., anchors, in each basic block of the encoder. Meanwhile, the point features of down-sampled points and anchors are also refined through the encoder. The anchors are further scattered into different 3D locations by learning specific offsets. Finally, AnchorFormer combines the down-sampled points and anchors as sparse points, and deliberately devises a morphing scheme to deform a canonical 2D grid at the location of each sparse point into a 3D structure. The whole architecture of AnchorFormer is optimized with respect to two objectives: the Chamfer Distance between the predicted points and the ground-truth points, and the compactness constraint of the generated points in each pattern.

The main contribution of this paper is the proposed AnchorFormer for reconstructing shape in point cloud completion. This issue also leads to the elegant views of how to characterize the geometric patterns in an object and how to convert the patterns into the fine-grained 3D structures. Extensive experiments over four datasets demonstrate the effectiveness of AnchorFormer from both quantitative and qualitative perspectives.

2. Related Work

Early works [2,6,18,26] on shape completion usually infer the missing parts based on hand-crafted features such as surface smoothness or symmetry axes. Other works [13,25] rely on large scale 3D object datasets and formulate the task as the matching of similar patches. With the development of deep learning, the most recent approaches formulate shape completion based on deep models. We categorize the related works in this direction into two groups, i.e., voxelization based and point cloud based shape completion.

Voxelization based shape completion. The success of convolution neural networks (CNN) in 2D image analysis prompted the application of 3D CNN to 3D data understanding. One natural solution for shape completion can be 3D voxel-level generation which is directly inspired by pixel-level image inpainting. Several works [10, 30, 38] study the problem based on purely volumetric data. For instance, Han *et al.* [10] propose to learn the multi-scale shape structure of the incomplete voxels via a 3D CNN. In addition, the voxel can be employed as an intermediate descriptor in shape completion and further converted to other 3D representation [4, 43]. For example, GRNet [43] leverages a differentiable gridding layer to capture voxel correlation and then converts the predicted voxels to point clouds. Despite the good model capacity of 3D CNN for feature learning, the geometric information loss caused by voxelization still makes it difficult for fine-grained reconstruction.

Point cloud based shape completion. The research [19,27,33,40,44,48,51] processing raw point cloud data for shape completion has largely proceeded along the scheme of point feature learning plus point generation. The works [45, 47] of exploring point feature utilize the MLP-based networks to learn a global feature vector of objects, and then predict the 3D structure based on that vector. To seek richer information for point cloud completion, the cascaded networks [11, 12, 31] are adopted to extract point features from different layers in a multi-scale manner. A representative work is PF-Net [12], which predicts the missing points in a hierarchical decoding scheme. Inspired by image [14] and video [16, 17, 24] Transformer, point feature learning via self-attention [46, 50] starts to emerge. Moreover, there exist other directions for point feature learning, such as adversarial learning [35] to make the generated shape realistic and cross-modality feature learning [8, 49] that facilitates shape reconstruction with the image. Point generation is the subsequent step after the learning of point features. One of the early work is FoldingNet [45] which maps the 2D grid onto a 3D surface to generate points through leveraging global feature of objects. Wen *et al.* [36] further upgrade the folding operation and introduce hierarchical folding to preserve 3D structures in different point resolution. More recently, Xiang *et al.* [41] devise a snowflake-like point growth method where child points are generated by splitting their parent points to capture local details. Nevertheless, these methods decode the shape from a global feature vector and may still suffer from the robustness problem in fine-grained pattern reconstruction.

In summary, our work belongs to point cloud based shape completion techniques. Different from the methods that employ a global feature vector for object reconstruction, our AnchorFormer contributes by investigating not only how to characterize geometric patterns through learning a set of anchors, but also how the anchors can be better leveraged for high-quality 3D shape reconstruction.

3. AnchorFormer

In this section, we present our proposal of AnchorFormer. Figure 2 shows an overview of our AnchorFormer

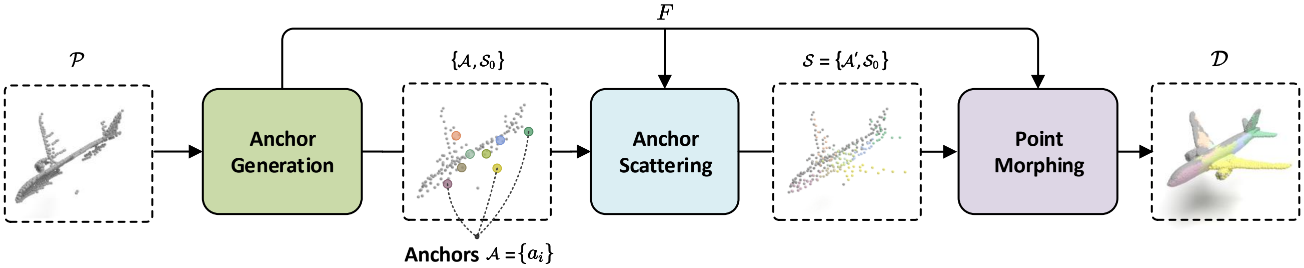

F
$\mathcal{P}$
Anchor Generation
$\{\mathcal{A}, \mathcal{S}_0\}$
Anchor Scattering
$\mathcal{S} = \{\mathcal{A}', \mathcal{S}_0\}$
Point Morphing
$\mathcal{D}$
Anchors $\mathcal{A} = \{a_i\}$

Figure 2. An overview of our AnchorFormer. Given the input partial observation $\mathcal{P}$ of an object, the points are down-sampled as $\mathcal{S}_0$ and the point features are extracted to predict a set of discriminative nodes $\mathcal{A}$, i.e., anchors, via the transformer encoder. The features F of the anchors and down-sampled points are learnt during feature encoding. Through learning specific offsets on F, the anchors are further scattered into different 3D locations. Taking the anchors $\mathcal{A}'$ after scattering and the down-sampled points $\mathcal{S}_0$ as sparse points $\mathcal{S}$, a morphing scheme is devised to reconstruct the detailed 3D structures at the location of each sparse point for dense points $\mathcal{D}$ prediction.

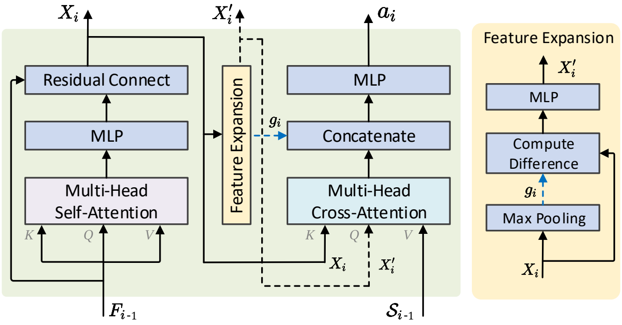


Figure 3. The detailed structure of the dual attention block.

architecture for point cloud completion. Specifically, AnchorFormer first down-samples the input points and extracts the point features via a feature extractor, followed by the prediction of a set of discriminative nodes, i.e., anchors, in transformer encoder. The features of the sampled points and anchors are refined during feature encoding. Next, the anchors are scattered into different 3D locations by learning particular offsets. Finally, AnchorFormer groups the down-sampled points and anchors as sparse points, and leverages a morphing scheme to convert a 2D grid at the location of each sparse point into a 3D structure. AnchorFormer is end-to-end trained by optimizing two objectives, i.e., the reconstruction of the predicted points, and the compactness constraint of the generated points in each pattern.

3.1. Anchor Generation

Most existing point cloud completion approaches encode the input partial observation as a global feature vector via a pooling operation, and then decode the vector for shape reconstruction. Nevertheless, the pooling operation may result in the loss of geometric details, and limits the capability of the global feature. Therefore, it is difficult to leverage such diluted global feature for reconstruction of various patterns in objects. To address this issue, we exploit the recipe of modeling object parts with key points [39] and introduce to learn a set of anchors to facilitate the reconstruction of local patterns. Particularly, we design the learning of anchors in two steps: feature extraction and anchor prediction.

Feature Extraction. Given the points $\mathcal{P}$ of the input partial observation, we first adopt an EdgeConv-based head [32] to down-sample the input observation into N points and extract the point features with dimension C. The down-sampled points $\mathcal{S}_0 \in \mathbb{R}^{N\times 3}$ and the corresponding point features $F_0 \in \mathbb{R}^{N\times C}$ are obtained by

$$\mathcal{S}_0 = FPS(\mathcal{P}), \;\; F_0 = Conv(\mathcal{P}, \mathcal{S}_0), \tag{1}$$

where $FPS(\cdot)$ denotes the farthest point sampling operation [21], and $Conv(\cdot)$ presents the EdgeConv-based networks. The output point features F_0 are further converted to a feature sequence and fed into a transformer encoder in our AnchorFormer for anchor prediction.

Anchor Prediction. Next, we devise a transformer encoder to jointly predict the anchor coordinates, and refine the features of the down-sampled points and the anchors. Specifically, a dual attention block is designed as the basic unit of our transformer encoder. The current dual attention block predicts a set of new anchors, and refines the input point features from the previous block at the same time. To characterize unobserved parts, a feature expansion module is presented to employ the feature difference between the input point features and the corresponding pooled feature vector for the anchor feature prediction. The cross-attention between the predicted anchor features and input point features is utilized for anchor coordinate learning.

Figure 3 depicts the structure of the basic dual attention block in our transformer encoder. For the i-th block, the input point features $F_{i-1} \in \mathbb{R}^{N_{i-1}\times C}$ of the N_{i-1} input points $\mathcal{S}_{i-1} \in \mathbb{R}^{N_{i-1}\times 3}$ are enhanced through the self-attention mechanism. We take the enhanced point features as $X_i \in \mathbb{R}^{N_{i-1}\times C}$ and exploit the feature expansion module to predict the features of L anchors. Through the linear projection on the feature difference between the enhanced point features X_i and the corresponding pooled feature vector g_i, the anchor features $X_i' \in \mathbb{R}^{L\times C}$ are learnt by

$$g_i = MaxPool(X_i), \;\; X_i' = MLP(g_i - X_i), \tag{2}$$

where $MaxPool(\cdot)$ and $MLP(\cdot)$ denote the max pooling

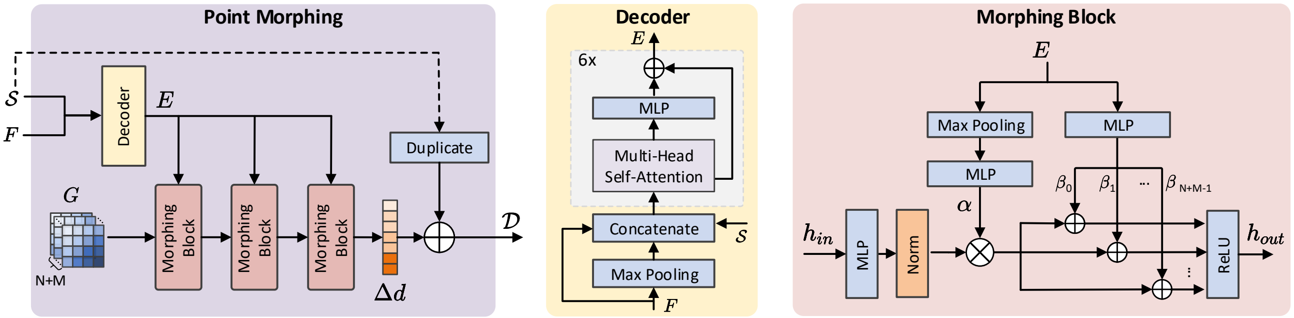


Figure 4. Illustration of the Point Morphing scheme for fine-grained pattern reconstruction. The structures of the transformer decoder (including six decoder blocks) and the morphing block are also detailed.

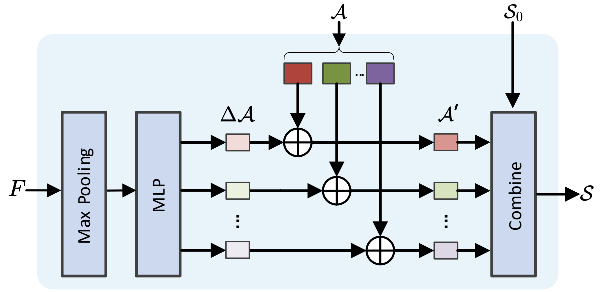


Figure 5. The detailed structure for Anchor Scattering.

operation and multilayer perceptron, respectively. We further compute the cross-attention weights between the enhanced point features $X_i \in \mathbb{R}^{N_{i-1} \times C}$ and the predicted anchor features $X_i' \in \mathbb{R}^{L \times C}$, and employ the weights to aggregate input points $\mathcal{S}_{i-1} \in \mathbb{R}^{N_{i-1} \times 3}$. The resultant aggregation is finally fused with the pooled feature vector g_i to predict the coordinates $a_i \in \mathbb{R}^{L \times 3}$ of L anchors:

$$a_i = MLP(Con[CroAtten(X_i, X_i', \mathcal{S}_{i-1}), g_i]), \qquad (3)$$

where $CroAtten(\cdot)$ and $Con[\cdot]$ are the cross-attention and feature concatenation. Note that the input point features and the input points for the first dual-attention block are F_0 and $\mathcal{S}_0$, which are extracted from the EdgeConv-based head. For the (i+1)-th dual attention block, we concatenate the enhanced point features X_i with the predicted anchor features X_i' as the input feature $F_i \in \mathbb{R}^{N_i \times C}$ ($N_i = N_{i-1} + L$). Similarly, we combine the input points $\mathcal{S}_{i-1}$ with the predicted anchors a_i as the input points $\mathcal{S}_i \in \mathbb{R}^{N_i \times 3}$ for (i+1)-th block. As such, the transformer encoder progressively increases the anchors and anchor features via cascaded dual attention blocks. We take all the predicted M anchors as $\mathcal{A} \in \mathbb{R}^{M \times 3}$ and the output point features from the last dual attention block as $F \in \mathbb{R}^{(N+M) \times C}$. The anchors $\mathcal{A}$, down-sampled points $\mathcal{S}_0$ and features F of all points are employed for the subsequent point generation.

3.2. Anchor Scattering

Given the learnt anchors from the transformer encoder and the down-sampled points of input observation, we aim to enrich the fine-grained details around those sparse points. Nevertheless, the anchors predicted by an encoder block during feature encoding often cluster in a local location, and thus it is hard to exploit these anchors to represent the holistic object shape. In other words, there are not enough anchors located in the space of the missing parts to facilitate the detailed structure reconstruction. To address it, we propose to scatter the anchors into different locations through learning specific offsets to capture different patterns.

We formally detail the formulation of anchor scattering in Figure 5. Given the point features $F \in \mathbb{R}^{(N+M) \times C}$ from the last encoder block of transformer encoder, we predict the anchor offsets $\Delta\mathcal{A} \in \mathbb{R}^{M \times 3}$ via a linear projection on the max pooled feature vector of F. Each anchor is then scattered into different 3D locations through adding the learnt offset, and the scattered anchors $\mathcal{A}' \in \mathbb{R}^{M \times 3}$ are obtained by

$$\Delta\mathcal{A} = MLP(MaxPool(F)), \;\; \mathcal{A}' = \mathcal{A} + \Delta\mathcal{A}. \qquad (4)$$

Through exploring the global shape information of the input points for offset prediction, the anchors are expected to be scattered into the space of the missing patterns for learning a holistic object shape. We take both of the scattered anchors $\mathcal{A}'$ and the down-sampled points $\mathcal{S}_0$ of input observation as the sparse points $\mathcal{S} \in \mathbb{R}^{(N+M) \times 3}$ for the following fine-grained 3D structure reconstruction.

3.3. Point Morphing

Point cloud completion advances [9,45] usually integrate the global feature vector into the deformation of a 2D grid for point generation. In view that solely exploiting the global feature vector to capture precise 3D details is insufficient, the relationship between the surrounding points should be also considered in the local pattern completion. By further incorporating the local point features into the reconstruction of detailed 3D structures, we propose a point morphing scheme to control the 2D grid deformation at the location of each sparse point.

Figure 4 depicts the pipeline of our point morphing scheme. Given the input sparse points $\mathcal{S}$ and the corresponding input point features F, a transformer decoder which consists of six decoder blocks is first utilized for feature fusion. We take the output feature from the transformer decoder as the decoded point features $E \in \mathbb{R}^{(N+M) \times C}$, which are then fed into three cascaded point morphing blocks for local pattern reconstruction. Specifically, as shown in the right part of Figure 4, we calculate the global feature vector $\alpha \in \mathbb{R}^{C_m}$ and local point features $\boldsymbol{\beta} \in \mathbb{R}^{(N+M)\times C_m} = \{\beta_j\}_{j=0}^{N+M-1}$ for all $N+M$ sparse points:

$$\alpha = MLP(MaxPool(E)), \;\; \boldsymbol{\beta} = MLP(E), \tag{5}$$

where the output feature dimension of MLP in the m-th morphing block is C_m. For the reconstruction of the pattern around the j-th sparse point, we leverage the global feature $\alpha \in \mathbb{R}^{C_m}$ and the local point feature $\beta_j \in \mathbb{R}^{C_m}$ as the affine parameters to modulate the 2D grid deformation. Given the input grid features $h_{in} \in \mathbb{R}^{K\times C_m}$ (K denotes point number of each grid), the output grid features h_{out} for the j-th sparse point are computed by

$$h_{out} = \alpha\frac{h_{in}-\mu}{\sigma} + \beta_j, \tag{6}$$

where μ and σ denotes the mean and standard deviation of a mini-batch of h_{in}. Note that the input grid features for the first morphing block are the canonical 2D grid $G \in \mathbb{R}^{K\times 2}$, and we set the output feature dimension C_m of the last morphing block as 3 to obtain the final output grid features, i.e., the 3D offsets $\Delta d_j \in \mathbb{R}^{K\times 3}$ for the j-th sparse point. Then, we duplicate the coordinates $s_j \in \mathbb{R}^3$ of the j-th sparse point K times and fuse them with the learnt 3D offsets to obtain the surrounding dense points $d_j \in \mathbb{R}^{K\times 3}$:

$$d_j = Dup(s_j) + \Delta d_j, \tag{7}$$

where $Dup(\cdot)$ denotes the point duplication. As such, each local pattern around one sparse point is described by K dense points. Finally, we collect all the dense points surrounding each sparse point to form the output dense points $\mathcal{D} \in \mathbb{R}^{[(N+M)\times K]\times 3}$ for our AnchorFormer.

3.4. Network Optimization

The architecture of AnchorFormer is end-to-end learnt by optimizing two objectives. One is the commonly adopted Chamfer Distance loss to minimize the distance between the predicted points and the ground truth. The other one is a compactness constraint to regulate the generated dense points in each fine-grained pattern.

We measure the point reconstruction via optimizing the L1 Chamfer Distances (CD_{L1}) [5] from two aspects: the distance between the predicted sparse points $\mathcal{S}$ and the ground truth $\mathcal{G}$, and the distance between the predicted dense points $\mathcal{D}$ and the ground truth $\mathcal{G}$. Therefore, the reconstruction loss L_{rec} is formulated as:

$$\mathcal{L}_{rec} = CD_{L1}(\mathcal{S},\mathcal{G}) + CD_{L1}(\mathcal{D},\mathcal{G}). \tag{8}$$

To facilitate the detailed structure reconstruction, we additionally employ a constraint [15] to guarantee the generated dense points surrounding a sparse point to be compact in each pattern. The loss function is to optimize the distance between local points based on a minimum spanning tree which is constructed on the point coordinate set $\mathcal{P}_r$. Such loss function $\mathcal{L}_{tree}$ is defined as:

$$\mathcal{L}_{tree}(\mathcal{P}_r,\lambda) = \sum_{(u,v)\in\mathcal{T}(\mathcal{P}_r)} \mathbb{I}\{Ed(u,v) \geq \lambda\epsilon\}Ed(u,v), \tag{9}$$

where $\mathcal{T}(\cdot)$ is the minimum spanning tree which is built on the predicted point coordinates $\mathcal{P}_r$. We denote $Ed(u,v)$ as the Euclidean distance between the vertex u and v in that tree. ϵ is the average edge length of the tree and $\mathbb{I}$ is the indicator function. We employ λ as a scale ratio to adjust the penalty of the distance. Thus, the proposed regularization term of point compactness $\mathcal{L}_{cpa}$ derived from $\mathcal{L}_{tree}$ is formulated as follows:

$$\mathcal{L}_{cpa} = \sum\nolimits_{j=0}^{N+M-1} \mathcal{L}_{tree}(d_j,\lambda), \tag{10}$$

where d_j denotes the predicted dense points around j-th sparse point as mentioned in Eq.(7) and λ is the scale ratio.

The overall training objective $\mathcal{L}$ in our AnchorFormer integrates the point reconstruction loss and the point compactness constraint:

$$\mathcal{L} = \mathcal{L}_{rec} + \gamma\mathcal{L}_{cpa}, \tag{11}$$

where γ is the trade-off parameter.

4. Experiments

4.1. Experimental Settings

Datasets. We empirically verify the merit of our AnchorFormer on PCN [47], ShapeNet-55/34 [46] and KITTI [7]. The **PCN** dataset consists of $28,974$ shapes for training and $1,200$ shapes for testing from 8 categories, which are sampled from the ShapeNet [3] dataset. The input partial observations are generated by back-projecting 2.5D depth images from 8 different views. The **ShapeNet-55/34** datasets are also derived from ShapeNet. We follow the standard protocols in [46, 50] to evaluate models on the two datasets. In particular, **ShapeNet-55** contains 55 categories and includes $41,952$ and $10,518$ shapes in the training and testing sets, respectively. The training data of **ShapeNet-34** are $46,765$ shapes from 34 categories, and the testing data of $5,705$ shapes are divided into two parts: $3,400$ shapes from 34 seen categories and $2,305$ shapes from 21 unseen classes. The evaluations on **ShapeNet-55/34** are conducted on the point cloud data masked with a ratio of 25%, 50% and 75%, accordingly formulating the completion task at three difficulty levels of simple (S), moderate (M) and hard (H). For the **KITTI** dataset, there are 2,401 partial car shapes extracted from outdoor 3D scenes. We use the standard setting in [43,46,50] to employ all the shapes in KITTI as the testing data, and the models for evaluation are trained on the subset of PCN which contains all car shapes.

Table 1. Performance comparison in terms of L1 Chamfer Distance $\times 10^3$ (CD_{L1}) on the PCN dataset. The Chamfer Distance performances of each category and the averaged result across all categories are all listed. (Lower CD_{L1} is better)

Method	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat	CD_{L1}
FoldingNet [45]	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31
TopNet [29]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15
PCN [47]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64
GRNet [43]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
PMPNet [37]	5.50	11.10	9.62	9.47	6.89	10.74	8.77	7.19	8.66
PoinTr [46]	4.05	9.34	7.97	7.92	6.40	9.29	6.66	6.47	7.26
SnowFlakeNet [41]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21
SeedFormer [50]	3.85	9.05	8.06	7.06	**5.21**	8.85	6.05	5.85	6.74
AnchorFormer	**3.70**	**8.94**	**7.57**	**7.05**	**5.21**	**8.40**	**6.03**	**5.81**	**6.59**

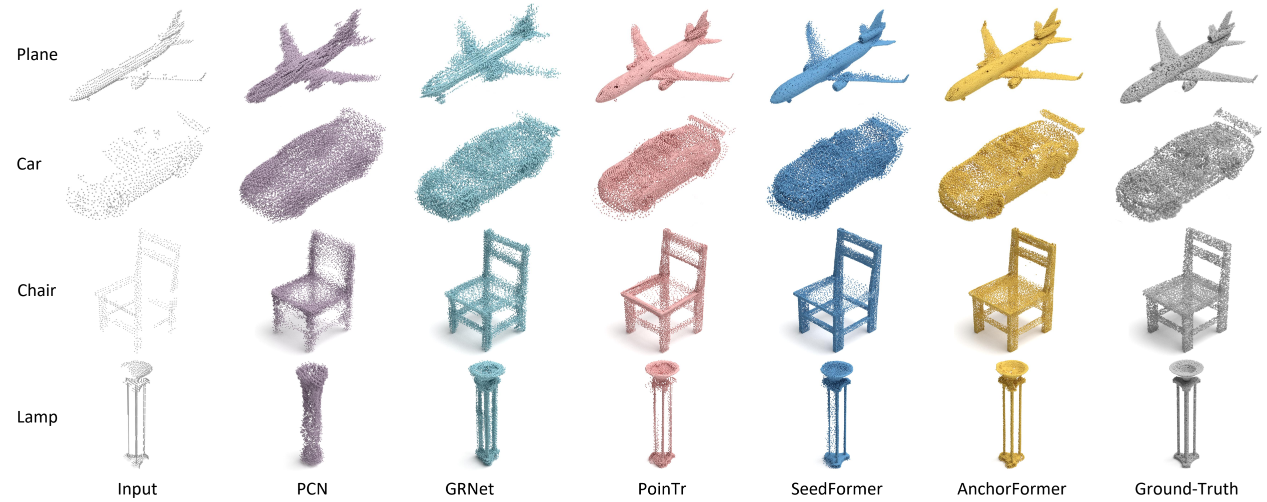


Figure 6. Four visual examples of point cloud completion results by different approaches on the PCN dataset. Different colors denote the point clouds reconstructed by different approaches.

Implementation Details. We implement our AnchorFormer on the PyTorch platform. The number N of the down-sampled points from EdgeConv-based head is 128. The encoder of AnchorFormer consists of 8 cascaded dual attention blocks, and the vanilla point transformer decoder proposed in [46] is adopted as our decoder. The number L of the predicted anchors in each dual attention block in transformer encoder is set as 16 and the total number M of the anchors is 128. The number K of the grid points is set as 64. The parameters of λ and γ are determined by cross validation and set as 1.2 and 0.05 empirically. Our networks are trained by exploiting AdamW optimizer with the base learning rate set as 0.0002.

Evaluation Metrics. We employ the L1/L2 Chamfer Distance and the F-Score [28] as the evaluation metrics for the PCN and ShapeNet-55/34 datasets. On KITTI, we follow [46, 50] to report the Fidelity Distance (FD) and Minimal Matching Distance (MMD) performances.

4.2. Comparisons with State-of-the-Art Methods

We compare our AnchorFormer with several state-of-the-art techniques, including FoldingNet [45], TopNet [29], PCN [47], GRNet [43], PMPNet [37], PoinTr [46], SnowFlakeNet [41] and SeedFormer [50], on the PCN, ShapeNet-55/34 and KITTI datasets.

Evaluation on PCN. Table 1 summarizes the L1 Chamfer Distance (CD_{L1}) comparisons on eight categories of the PCN dataset. AnchorFormer consistently outperforms all baselines in terms of both per-category Chamfer Distance and the averaged distance. In general, lower Chamfer Distance indicates more accurate reconstructive shape. Specifically, AnchorFormer achieves the averaged CD_{L1} of 6.59, which reduces the Chamfer Distance of the best competitor SeedFormer by 0.15. Though both of SeedFormer and AnchorFormer rebuild the 3D shape from a set of key points, they are fundamentally different in that SeedFormer estimates the seed features through interpolating the features of the observed partial points, and AnchorFormer dynamically refines anchor features in transformer encoder to better capture local geometry. As indicated by the results, learning more powerful pattern features does benefit shape reconstruction. Figure 6 further visualizes the point cloud completion results of four different shapes. In particular, AnchorFormer predicts high-quality object shapes with smoother surfaces (e.g., the body of the airplane) and more fine-grained local structures (e.g., the wheels of the car). In addition, there is less noise in the point clouds generated by our AnchorFormer. The results demonstrate the advan-

Table 2. Performance comparison in terms of L2 Chamfer Distance $\times 10^3$ (CD$_{L2}$) and F-Score@1% (F1) on the ShapeNet-55 dataset. The per-category L2 Chamfer Distance results are reported on 5 categories with most training samples and 5 categories with the least training samples. CD$_{L2}$-S, CD$_{L2}$-M and CD$_{L2}$-H denote the L2 Chamfer Distance on the masked point cloud with the ratio of 25%, 50% and 75%, respectively. CD$_{L2}$ and F1 are the averaged results on all categories and all difficulties. (Lower CD$_{L2}$ and higher F1 are better)

Method	Table	Chair	Plane	Car	Sofa	Birdhouse	Bag	Remote	Keyboard	Rocket	CD$_{L2}$-S	CD$_{L2}$-M	CD$_{L2}$-H	CD$_{L2}$	F1
FoldingNet [45]	2.53	2.81	1.43	1.98	2.48	4.71	2.79	1.44	1.24	1.48	2.67	2.66	4.05	3.12	0.082
TopNet [29]	2.21	2.53	1.14	2.18	2.36	4.83	2.93	1.49	0.95	1.32	2.26	2.16	4.30	2.91	0.126
PCN [47]	2.13	2.29	1.02	1.85	2.06	4.50	2.86	1.33	0.89	1.32	1.94	1.96	4.08	2.66	0.133
GRNet [43]	1.63	1.88	1.02	1.64	1.72	2.97	2.06	1.09	0.89	1.03	1.35	1.71	2.85	1.97	0.238
PoinTr [46]	0.81	0.95	0.44	0.91	0.79	1.86	0.93	0.53	0.38	0.57	0.58	0.88	1.79	1.09	0.464
SeedFormer [50]	0.72	0.81	0.40	0.89	0.71	1.51	0.79	0.46	0.36	0.50	0.50	0.77	1.49	0.92	0.472
AnchorFormer	**0.58**	**0.67**	**0.33**	**0.69**	**0.58**	**1.35**	**0.64**	**0.36**	**0.27**	**0.42**	**0.41**	**0.61**	**1.26**	**0.76**	**0.558**

Table 3. Performance comparison in terms of L2 Chamfer Distance $\times 10^3$ (CD$_{L2}$) and F-Score@1% (F1) on the ShapeNet-34 dataset. The L2 Chamfer Distance performances on both of the 34 seen categories and 21 unseen categories are reported. CD$_{L2}$-S, CD$_{L2}$-M and CD$_{L2}$-H denote the L2 Chamfer Distance on the masked point cloud with a ratio of 25%, 50% and 75%, respectively. CD$_{L2}$ and F1 are the averaged results on corresponding category subset (seen/unseen) across all difficulties. (Lower CD$_{L2}$ and higher F1 are better)

Method	34 seen categories					21 unseen categories				
	CD$_{L2}$-S	CD$_{L2}$-M	CD$_{L2}$-H	CD$_{L2}$	F1	CD$_{L2}$-S	CD$_{L2}$-M	CD$_{L2}$-H	CD$_{L2}$	F1
FoldingNet [45]	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
TopNet [29]	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PCN [47]	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
GRNet [43]	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
PoinTr [46]	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
SeedFormer [50]	0.48	0.70	1.30	0.83	0.452	0.61	1.07	2.35	1.34	0.402
AnchorFormer	**0.41**	**0.57**	**1.12**	**0.70**	**0.564**	**0.52**	**0.90**	**2.16**	**1.19**	**0.535**

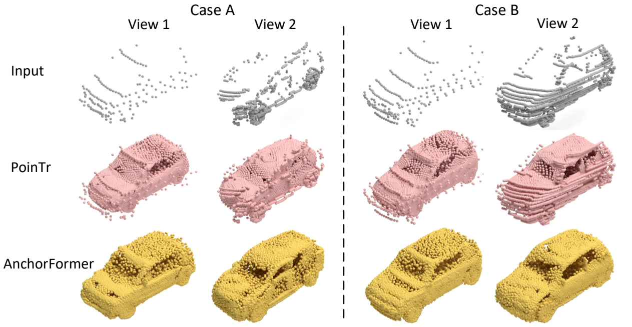


Figure 7. Point cloud completion results of two car shapes in two different views in the KITTI dataset.

Table 4. Fidelity Distance (FD) and Minimal Matching Distance (MMD) on KITTI. (Lower FD and MMD are better)

	TopNet [29]	PCN [47]	GRNet [43]	PoinTr [46]	AnchorFormer
FD	5.354	2.235	0.816	**0.000**	**0.000**
MMD	0.636	1.366	0.568	0.526	**0.458**

tage of characterizing regional information through learning a set of anchors to enhance point cloud completion.

Evaluation on ShapeNet-55. We then evaluate AnchorFormer on the ShapeNet-55 dataset with more categories. Table 2 lists the performances of L2 Chamfer Distance (CD$_{L2}$) of different approaches. In detail, we report the averaged CD$_{L2}$ and F1 values on all the categories, and the CD$_{L2}$ performances on the masked point cloud data with three different ratios, i.e., CD$_{L2}$-S, CD$_{L2}$-M and CD$_{L2}$-H. Moreover, we select and show the per-category CD$_{L2}$ of the five categories (Table, Chair, Plane, Car and Sofa) with the most training samples (>2,500), and the five categories (Birdhouse, Bag, Remote, Keyboard and Rocket) with the least training examples (<80). On all the experimental settings, our AnchorFormer leads to higher performances against other methods. In the case of training the model for the five categories with few data, AnchorFormer still exhibits improvements over SeedFormer, verifying the good model capacity to capture 3D shape information. AnchorFormer also surpasses SeedFormer by 0.086 in F1 score and the result indicates that AnchorFormer reconstructs the 3D shape with a higher percentage of the correct points.

Evaluation on ShapeNet-34. Following [46], we examine the generalization ability of AnchorFormer for novel object shape completion on ShapeNet-34. Table 3 details the CD$_{L2}$ on both the seen categories and unseen classes. As expected, the average performances on unseen categories are inferior to those on the seen categories. Despite having large shape differences between training data and testing unseen data, AnchorFormer still achieves 0.535 F1 score on 21 unseen categories and obtains 0.133 F1 gain over SeedFormer. The results basically validate the generalization ability of AnchorFormer for novel object reconstruction.

Evaluation on KITTI. Next, we also experiment with our AnchorFormer on KITTI as in [46] to test point cloud completion on real 3D car shape. Table 4 shows the Fidelity Distance (FD) and Minimal Matching Distance (MMD) of

Table 5. Performance comparisons among different variants of AnchorFormer on the PCN dataset.

Model	Anchor	Morphing	L_{cpa}	CD_{L_1}	F1
A	Global Feature	Folding	-	7.33	0.792
B	√	Folding	-	6.81	0.810
C	√	Style-based Folding	-	6.77	0.814
D	√	√	-	6.68	0.820
E	√	√	√	**6.59**	**0.827**

different methods. AnchorFormer constantly performs better than other models with respect to both two metrics. On one hand, the lowest FD attained by AnchorFormer reflects that the input structure is well preserved with shape reconstruction. On the other hand, the lowest MMD indicates that the shape predicted by AnchorFormer is more like a car than other approaches. Furthermore, Figure 7 showcases point cloud completion of two examples in two views. AnchorFormer recreates the shape with better quality in fine-grained patterns, which manifests the merit of leveraging anchors to reform the detailed 3D structures.

4.3. Analysis of AnchorFormer

Model Design. Here, we study how each design in our AnchorFormer impacts the overall performance of point cloud completion. Table 5 lists the performance comparisons among different variants of AnchorFormer. We start from the basic model (**A**), which leverages a vanilla transformer encoder [46] to learn the global feature vector of an object and then decodes the vector via a transformer decoder, following by the folding operation [45] to generate the points. The model **B** upgrades the basic model A through learning a set of discriminative nodes, i.e., anchors, to characterize regional information, and improves F1 score from 0.792 to 0.81. The model **C** and **D** further replace the folding operation with Style-based Folding [42] and our point morphing scheme, respectively. Compared to Style-based Folding that only integrates the global feature vector of an object into the 2D grid deformation procedure, point morphing jointly adjusts the deformation by both of the local point features and the global object feature for fine-grained pattern reconstruction. As such, the model D attains better CD_{L1} and F1 score than the model C. Finally, the model **E**, i.e., our AnchorFormer, by regulating the compactness of the generated points in each pattern, shows the best performances.

Visualization Analysis. To better qualitatively verify the effectiveness of completing point cloud from anchors, we further visualize the formation of anchors, sparse points, and dense points in Figure 8. Note that we plot the sparse points and dense points which are derive from the identical anchor in the same color. As shown in the figure, the anchors distribute at both the observed (e.g., the body of the boat in the third example) and unobserved (e.g., the seat and back of the sofa in the first case) locations. Through point scattering operation, the anchors are then scattered around each location and combined with the down-sampled points of the input observation as sparse points, to rebuild a coarse 3D structure of an object. The fine-grained structure at each sparse point is further reformed by the morphing scheme. The results indicate that AnchorFormer benefits from the learning of a set of anchors, and enriches the details of the local pattern around each sparse point, leading to a completed 3D object shape.

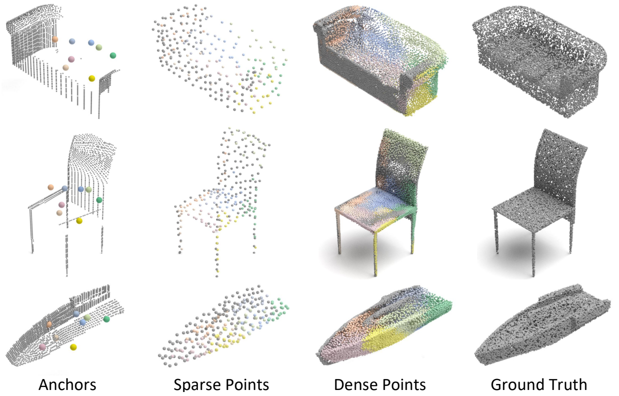


Figure 8. Visualization of the formation of anchors, sparse points, and dense points for three shapes from the PCN dataset with their corresponding ground truth. We plot the sparse points and dense points that are derived from the identical anchor in the same color.

5. Conclusions

We have presented AnchorFormer that explores the regional discrimination for point cloud completion. In particular, we study the problem of completing object shape from learning a set of discriminative nodes, i.e., anchors, to characterize different local geometric patterns. To materialize our idea, AnchorFormer first predicts a series of anchors from the input partial observation via the transformer encoder. Through learning specific offsets, the anchors are further scattered into different 3D locations and combined with the down-sampled points of the input observation as sparse points. Finally, a point morphing scheme is deliberately designed to reconstruct the fine-grained 3D structure at the location of each sparse point by deforming a canonical 2D grid. Experiments demonstrate the superiority of our AnchorFormer and qualitative evaluations on point cloud completion also show that it nicely models the fine-grained geometry and reconstructs the missing parts precisely.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Contract 62021001. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution and the Supercomputing Center of the USTC.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning Representations and Generative Models for 3D Point Clouds. In *ICML*, 2018. 1

[2] 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 *Eurographics*, 2014. 2

[3] Angel X. Chang, Thomas Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[4] Angela Dai, Charles R. Qi, and Matthias Nießner. Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis. In *CVPR*, 2017. 2

[5] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In *CVPR*, 2017. 5

[6] Zeqing Fu, Wei Hu, and Zongming Guo. Local Frequency Interpretation and Non-Local Self-Similarity on Graph for Point Cloud Inpainting. *IEEE TIP*, 2019. 2

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *IJRR*, 2013. 5

[8] Bingchen Gong, Yinyu Nie, Yiqun Lin, Xiaoguang Han, and Yizhou Yu. ME-PCN: Point Completion Conditioned on Mask Emptiness. In *ICCV*, 2021. 2

[9] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018. 4

[10] 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 *ICCV*, 2017. 2

[11] Tianxin Huang, Hao Zou, Jinhao Cui, Xuemeng Yang, Mengmeng Wang, Xiangrui Zhao, Jiangning Zhang, Yi Yuan, Yifan Xu, and Yong Liu. RFNet: Recurrent Forward Network for Dense Point Cloud Completion. In *ICCV*, 2021. 2

[12] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In *CVPR*, 2020. 2

[13] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A Probabilistic Model for Component-Based Shape Synthesis. *ACM TOG*, 2012. 2

[14] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual Transformer Networks for Visual Recognition. *IEEE TPAMI*, 2022. 2

[15] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and Sampling Network for Dense Point Cloud Completion. In *AAAI*, 2020. 5

[16] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Jiebo Luo, and Tao Mei. Stand-Alone Inter-Frame Attention in Video Models. In *CVPR*, 2022. 2

[17] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. Dynamic Temporal Filtering in Video Models. In *ECCV*, 2022. 2

[18] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A Field Model for Repairing 3D Shapes. In *CVPR*, 2016. 2

[19] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational Relational Point Completion Network. In *CVPR*, 2021. 2

[20] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *ICCV*, 2019. 1

[21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 1, 3

[22] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 1

[23] Zhaofan Qiu, Yehao Li, Yu Wang, Yingwei Pan, Ting Yao, and Tao Mei. SPE-Net: Boosting Point Cloud Analysis via Rotation Robustness Enhancement. In *ECCV*, 2022. 1

[24] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, and Tao Mei. MLP-3D: A MLP-like 3D Architecture with Grouped Time Mixing. In *CVPR*, 2022. 2

[25] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure Recovery by Part Assembly. *ACM TOG*, 2012. 2

[26] Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas J. Guibas. Data-Driven Structural Priors for Shape Completion. *ACM TOG*, 2015. 2

[27] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. LAKe-Net: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints. In *CVPR*, 2022. 2

[28] Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What Do Single-view 3D Reconstruction Networks Learn? In *CVPR*, 2019. 6

[29] Lyne P. Tchapmi, Vineet Kosaraju, S. Hamid Rezatofighi, Ian Reid, and Silvio Savarese. TopNet: Structural Point Cloud Decoder. In *CVPR*, 2019. 6, 7

[30] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: A Patch-based Deep Representation of 3D Shapes. *ACM TOG*, 2018. 2

[31] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded Refinement Network for Point Cloud Completion. In *CVPR*, 2020. 2

[32] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM TOG*, 2019. 2, 3

[33] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. SoftPoolNet: Shape Descriptor for Point Cloud Completion and Classification. In *ECCV*, 2020. 2

[34] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Learning Local Displacements for Point Cloud Completion. In *CVPR*, 2022. 1

[35] 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 *CVPR*, 2021. 2

[36] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point Cloud Completion by Skip-attention Network with Hierarchical Folding. In *CVPR*, 2020. 2

[37] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths. In *CVPR*, 2021. 6

[38] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning Shape Priors for Single-View 3D Completion and Reconstruction. In *ECCV*, 2018. 2

[39] Weikun Wu, Yan Zhang, David Wang, and Yunqi Lei. SK-Net: Deep Learning on Point Cloud via End-to-end Discovery of Spatial Keypoints. In *AAAI*, 2020. 3

[40] Yaqi Xia, Yan Xia, Wei Li, Rui Song, Kailang Cao, and Uwe Stilla. ASFM-Net: Asymmetrical Siamese Feature Matching Network for Point Completion. In *ACM MM*, 2021. 2

[41] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer. In *ICCV*, 2021. 1, 2, 6

[42] 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 *CVPR*, 2021. 8

[43] 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. 2, 5, 6, 7

[44] Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. ShapeFormer: Transformer-based Shape Completion via Sparse Representation. In *CVPR*, 2022. 2

[45] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. In *CVPR*, 2018. 2, 4, 6, 7, 8

[46] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers. In *ICCV*, 2021. 1, 2, 5, 6, 7, 8

[47] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point Completion Network. In *3DV*, 2018. 1, 2, 5, 6, 7

[48] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail P-reserved Point Cloud Completion via Separated Feature Aggregation. In *ECCV*, 2020. 2

[49] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-Guided Point Cloud Completion. In *CVPR*, 2021. 2

[50] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. SeedFormer: Patch Seeds based Point Cloud Completion with Upsample Transformer. In *ECCV*, 2022. 2, 5, 6, 7

[51] Linqi Zhou, Yilun Du, and Jiajun Wu. 3D Shape Generation and Completion through Point-Voxel Diffusion. In *ICCV*, 2021. 2