This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

WalkFormer: Point Cloud Completion via Guided Walks

Mohang Zhang[1,2] Yushi Li[1*] Rong Chen[3] Yushan Pan[1]
Jia Wang[1] Yunzhe Wang[4] Rong Xiang[5]

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University
[2]University of Liverpool [3]Dalian Maritime University
[4]Suzhou University of Science and Technology [5]The Hong Kong Polytechnic University

Abstract

Point clouds are often sparse and incomplete in real-world scenarios. The prevailing methods for point cloud completion typically rely on encoding the partial points and then decoding complete points from a global feature vector, which might lose the existing patterns and elaborate structures. To address these issues, we propose WalkFormer, a novel approach to predict complete point clouds through a partial deformation process. Concretely, our method samples locally dominant points based on feature similarity and moves the points to form the missing part. Since these points maintain representative information of the surrounding structures, they are appropriately selected as the starting points for multiple guided walks. Furthermore, we design a Route Transformer module to exploit and aggregate the walk information with topological relations. These guided walks facilitate the learning of long-range dependencies for predicting shape deformation. Qualitative and quantitative evaluations demonstrate that our proposed approach achieves superior performance compared to state-of-the-art methods in the 3D point cloud completion task.

1. Introduction

The advancement of laser scanners and depth cameras has led to the widespread of 3D point cloud data, to describe real-world objects flexibly and conveniently. However, occlusion, transparency, and limitation in sensor resolution often result in the acquisition of incomplete point clouds [4]. The completed shapes have significant values that are essential for downstream tasks [6, 7, 13, 21, 22], and thus recovering complete 3D models from partial shapes remains an issue.

To predict complete shapes, PCN [48] is the first method that directly operates on the raw point clouds. Building upon PCN, some efforts [25, 32, 34, 39, 47] were focused on the encoder-decoder architecture in a coarse-to-fine manner. These methods encode incomplete input into a global feature vector and subsequently decode it to a coarse point cloud that represents the overall structure for further refinement. However, the challenge lies in effectively decoding the holistic shape along with fine-grained details, as the information loss during the max-pooling operation used for feature extraction. SoftPoolNet [31] introduced a soft-pooling operation to mitigate this issue by selecting multiple values instead of solely dependent on the maximum value. More recent work, such as FBNet [42], takes feedback features from high-level stages to refine the output recurrently. These generative models neglect the transformation in 3D space, which are deficient in detail generation and inevitably suffer from the aforementioned problems.

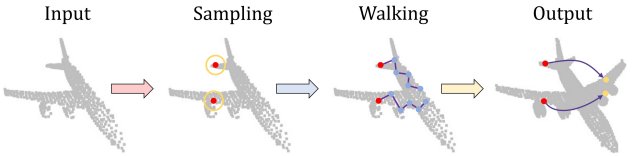

Figure 1. Illustration of the completion process. Our WalkFormer can adaptively sample the points from partial input to complete the point cloud. Meanwhile, a set of walks (blue points) on the point cloud provides rich information to guide the sampled points (red points) moving towards the missing regions (yellow points).

Some deformation-based methods [35, 36, 46] propose a different approach to overcome this difficulty. Rather than transforming 2D grids into 3D point clouds, these methods focus on a deformation process that occurs between 3D shapes. PMP-Net [35] and PMP-Net++ [36] predict a set of displacement vectors that describe the movement from an incomplete point cloud to the target one. Nevertheless, they move every point in the source input during each deformation step and thus often fail to preserve known structures. Additionally, the popular feature learning schemes for point

*Corresponding author: Yushi.Li@xjtlu.edu.cn

clouds are based on local spatial relationships, points in a local region will share similar features. Although P2P-NET [46] introduces noise augmentation for a richer variety of transformations to make the points leave their original positions, the uncertainty in noise and lack of long-range dependencies also limit their ability to predict accurate long displacements. Consequently, this results in sparse points in the missing regions and uneven distribution.

In this paper, we present a novel deformation-based architecture for point cloud completion, namely WalkFormer, which moves only the partial points in each deformation step. Specifically, our proposed method first adopts a Neighbour Similarity Sampling module to down-sample the points gathered in a local region based on their feature similarity as shown in Figure 1. These sampled points will be moved in the subsequent deformation step without losing the initial structures. Motivated by the success of nonlocal mechanism [29,40,43], WalkFormer conducts guided walks that contain sequences of points initiated from the sampled points to enrich feature diversity by long-range topology correlations, enabling the model to predict point displacements at longer distances. In addition, we propose a Route Transformer module, which takes topology-aware walk features into consideration to get a precise and consistent moving path. With this deformation approach and learned walk features, WalkFormer is capable of uniformly recovering the missing parts and preserving the existing structures. The main contributions of our work can be summarized as follows:

- We propose WalkFormer, an end-to-end model that improves the performance of point cloud completion through a multi-step partial deformation process.
- We introduce a new sampling method that ensures detail preservation in the original point cloud. With this strategy, our model is encouraged to focus on the representative structure information.
- We design the Route Transformer, which effectively aggregates long-range walk features across deformation steps to guide the points moving. In this module, we take previous displacements into current step and relate them via a topology-aware transformer module.

2. Related Work

Point Cloud Learning. Learning-based point cloud analysis has been widely studied in recent years, many applications successfully take 3D coordinates as inputs for a wide range of tasks such as shape classification [7], semantic segmentation [13], and point cloud registration [6].

PointNet [21] and PointNet++ [22] first use symmetrical operations to extract features on point clouds without voxelization. Later, a number of approaches take advantage of the 3D convolutions for feature learning. PointCNN [12] uses a χ-conv that automatically adapts the convolution kernel to 3D point clouds. RS-CNN [15] defines a continuous convolution to process point clouds. ConvPoint [1] learns the convolutional function by processing partial structures selected from a spatial sphere. As the graph is a natural way to represent the neighbourhood of each point, Wang *et al*. [30] propose DGCNN that dynamically updates the graph with local points. In addition, [10, 11] associate spectral-based GCN with GAN for accurate point cloud generation.

Unlike the convolution-based or graph-based methods that are proposed for local feature learning, PointASNL [43] exploits the point nonlocal cell to query nonlocal points in the entire point cloud. Moreover, CurveNet [40] groups sequences of points on the point cloud for long-range dependencies learning. GraphWalks [20] proposes an autoregressive model aimed at selecting vertices to approximate the shortest path on both mesh and point cloud. In our work, long-range features are aggregated by guided walks to facilitate the point completion task.

Point Cloud Completion. Early attempts [3, 23, 37] in shape completion simply migrated mature 2D completion methods (voxelization and convolution) to 3D space, resulting in time-consuming computation and high memory usage. Therefore, some methods [17, 24, 26, 49] exploit point representation learning and directly take raw point clouds as inputs to generate complete shapes. For instance, FoldingNet [45] performs the folding operation to deform a 2D grid lattice onto a 3D surface in an auto-encoder architecture. SA-Net [34] proposes hierarchical folding to generate point clouds with regional structure details progressively. On the other hand, PMP-Net [35] and PMP-Net++ [36] consider the deformation that occurs between 3D point clouds, from the partial input to the complete one. Besides, generative models such as GAN and VAE, are also adopted in point generation. PF-Net [8] produces the missing structures by integrating multi-stage completion with adversarial loss. Wang *et al*. [27, 28] design a generator with cascaded refinement to synthesize high fidelity objects. AutoSDF [18] relies on Vector-Quantized VAE to model 3D shape as a non-sequential autoregressive distribution.

With the transformer architecture becoming successful in point cloud learning [7, 38, 50], Yu *et al*. [47] formulate the completion task as a set-to-set translation problem. Similarly, SnowflakeNet [39] employs a skip-transformer to decode the growth procedure of point clouds. Based on upsample transformer, SeedFormer [51] stores regional features into patch seed for detailed shape recovery. Different from the methods that directly reconstruct a set of points, our proposed WalkFormer integrates the transformer structure into the partial deformation approach, specifically focusing on predicting the point displacements in 3D space.

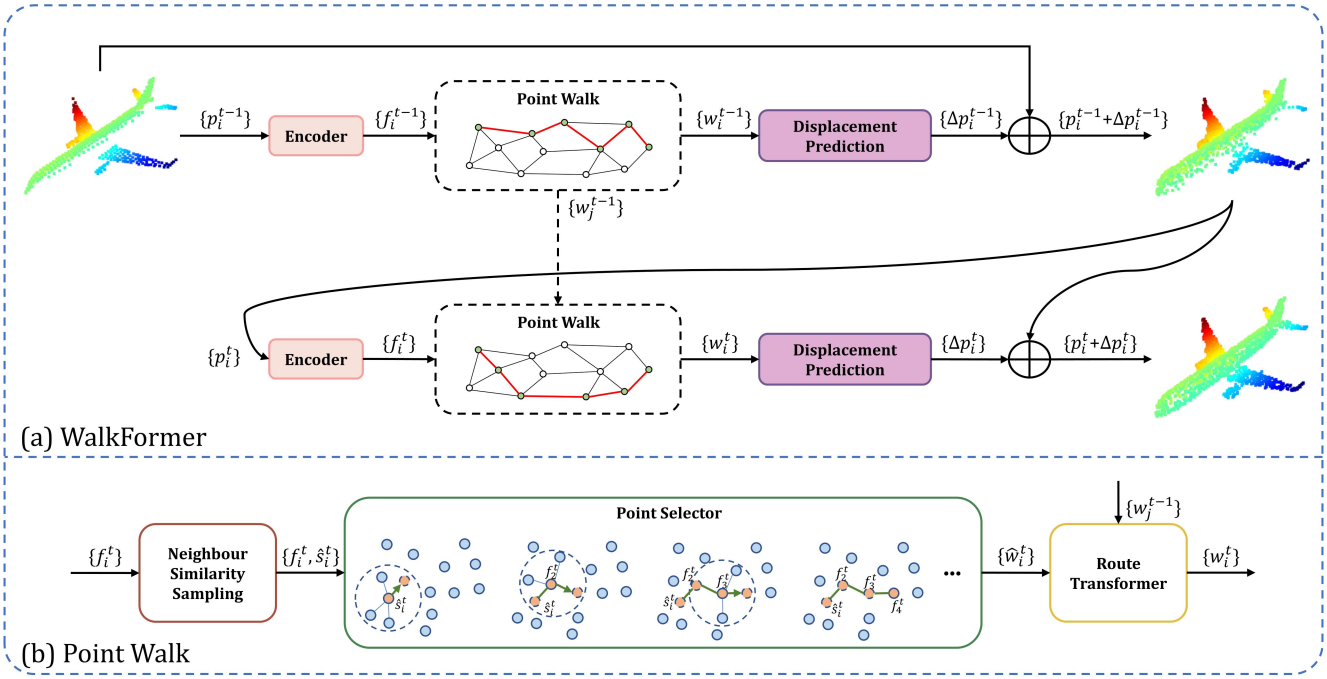

Figure 2. The overall architecture of our WalkFormer framework. (a) In each deformation step, the Encoder extracts point-wise features from the partial input point cloud. Subsequently, walk features from the Point Walk are applied to predict a set of displacement vectors from the original input to a complete point cloud. (b) The Point Walk module initiates by sampling several points as starting points to construct walks in the feature space. The walk features are then refined by Route Transformer, which leverages information from both the current and previous steps.

3. Method

3.1. Overview

The pipeline of WalkFomer is illustrated in Figure 2(a). In this section, we introduce the proposed WalkFomer for point cloud completion in detail.

Encoder. Given a partial point cloud set P, denoted as $P = \{p_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$, where each point is represented by its 3D coordinate, the WalkFormer stacks several set abstraction and feature propagation layers from [22] to extract per-point features. Moreover, point transformer blocks [50] are employed to exchange information within localized features. Consequently, a set of per-point features is obtained, denoted as $F = \{f_i\}_{i=1}^{N} \in \mathbb{R}^{N\times C}$, where C is the feature dimension.

Deformation-based Architecture. We implement the proposed WalkFormer in a deformation-based approach inspired by PMP-Net [35, 36]. It aims to complete the partial point cloud through a multi-step point moving procedure with a coarse-to-fine searching radius. The network prediction is a set of displacement vectors $\Delta P^t = \{\Delta p_i^t\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$ for total T steps, such that the complete point cloud $P' = \{p'_j\}_{j=1}^{N} \in \mathbb{R}^{N\times 3}$ is produced by $\{p'_j\} = \{p_i + \sum_{t=1}^{T} \Delta p_i^t\}$. Rather than moving all points at each step, our WalkFormer only samples a subset of N_m points for moving, also denoted as the starting points $P_s^t = \{p_i^t\}_{i=1}^{N_m} \in \mathbb{R}^{N_m\times 3}$, to construct walks in the point feature space F. These walks aid in inferring the moving track of the starting points during the deformation process. In each deformation step, the positions of the starting points are updated by adding the coordinate offsets obtained from the Displacement Prediction module $\{p_i^t\} = \{p_i^{t-1} + \Delta p_i^{t-1}\}$. This iterative update scheme allows for the generation of a complete point cloud.

3.2. Point Walk

As shown in Figure 2(b), the point walk module is devised to generate multiple walks that connect different points with informative features. We ignore the superscript t in the same step for convenience. Given the per-point features F extracted from the Encoder layer, a directed graph $G = (F, E)$ is constructed, where $F = \{f_1, f_2, ..., f_N\}$ and $E \subseteq V \times V$ denotes the set of edges. The edges are constructed using the k-nearest neighbours (K-NN) on coordinates P. By leveraging extracted features, the point walk module aims to discover walks on this graph that capture the topology relationship and establish long-range dependencies for larger receptive fields, facilitating our model to

better predict the spatial deformation.

Neighbour Similarity Sampling. The initial step of walk construction needs appropriate starting points. However, existing sampling algorithms such as farthest point sampling [22] and minimum density sampling [14] are prone to select points from the outer regions that possess distinctive geometric information, moving these points may discard the existing structures and destroy the inherent property of the source point cloud. To overcome these challenges, we propose Neighbour Similarity Sampling to downsample a set of N_m points as the starting points for the walks. Given the input point cloud, we first employ farthest point sampling (FPS) to obtain a relatively uniform set of sampled points $\{p_i\}_{i=1}^{N_m} \in \mathbb{R}^{N_m \times 3}$. Treating these points as centroids, we group K points within a radius of the centroids by the ball query algorithm. For each group of points $\{p_{i,1}, ..., p_{i,k}, ..., p_{i,K}\}$, we query their respective features $\{f_{i,1}, ..., f_{i,k}, ..., f_{i,K}\}$ to calculate pairwise affinities based on cosine similarity and take the point with largest affinity:

$$p_{i,k} = \arg\max_{k} \sum_{j=1}^{K} \frac{\langle f_{i,k}, f_{i,j} \rangle}{\|f_{i,k}\|_2 \|f_{i,j}\|_2}. \tag{1}$$

Here, $\|\cdot\|_2$ denotes L2 norm, $\langle \cdot, \cdot \rangle$ denotes inner product, and $p_{i,k}$ is sampled as the starting point. Hence, points with similar features to their neighbours will be sampled. These points maintain the dominant information of local structures which are suitable for the starting points.

Point Selector. A walk (of length l) in the graph G is defined as a non-empty sequence of vertices, such that $w = \{f_1, ..., f_l\} \in \mathbb{R}^{l \times C}$, where each vertex f_i is connected to its adjacent vertex f_{i+1}. Following the practice of [20, 40], we employ a policy $\pi(\cdot)$ that guides the selection of the next point during the walk:

$$f_{i+1} = \pi(f_i), 1 <= i <= l-1. \tag{2}$$

Specifically, the policy $\pi(\cdot)$ determines how to select the next point based on a selection logits α. Take the current point feature f_i, we calculate the selection logits on neighbouring point features f_j by using the attention mechanism:

$$\alpha_i = \frac{\beta(f_i)^T \gamma(f_j)}{\sqrt{d_k}}, j \in \mathcal{N}(i). \tag{3}$$

Here, $\mathcal{N}(i)$ denotes the k-nearest neighbours, β and γ are linear mapping functions and d_k is the dimension of the input features. The policy will then select the highest score $\varphi(\alpha_i)$ among K neighbours:

$$\pi(f_i) = \sum_{1}^{K} (\varphi(\alpha_i) \cdot \mathcal{N}(i)), \tag{4}$$

where $\cdot$ denotes broadcast multiplication and φ is gumbel-softmax [9, 16, 44] to replace the standard softmax which activates effective gradient flow. Relying on the constructed walks, our model is allowed to exploit the topology information in long-range regions.

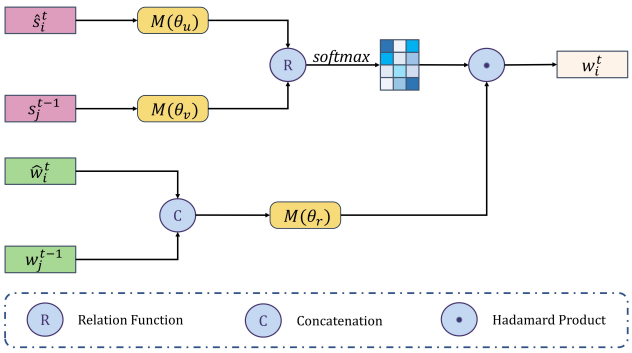

Figure 3. Detailed structure of the Route Transformer.

Route Transformer. Point cloud completion in a coarse-to-fine manner commonly fuses features from different stages to refine the current features. Among these methods [19, 39, 42], they often take the entire point cloud into account and overlook the details. Differently, in our method, the generated walk in each step contains a sequential arrangement of points. Furthermore, every moving path counts throughout the whole route, previous moving information is needed to decide the next move. Therefore, we introduce the Route Transformer that incorporates walk features from the previous step into the current step, which allows for predicting a more consistent moving path.

Specifically, Route Transformer update the intermediate walk features $\hat{w}_i^t = \{\hat{x}_i^t, \hat{x}_{q,1}^t, \hat{x}_{q,2}^t, ..., \hat{x}_{q,l-1}^t\} \in \mathbb{R}^{l \times C}$ from Point Selector module to $w_i^t = \{x_i^t, x_{q,1}^t, x_{q,2}^t, ..., x_{q,l-1}^t\} \in \mathbb{R}^{l \times C}$. We denote the starting point features $\hat{x}_i^t, x_i^t$ as $\hat{s}_i^t, s_i^t$. As described in Figure 3, we first use the current starting point features $\hat{s}_i^t$ as queries and s_j^{t-1} from the previous step as keys for cross attention, and then Route Transformer concatenates the walk features $\hat{w}_i^t$ with w_j^{t-1} to generate the values v_{ij}^t. To better focus on the points with similar topology structures, attention score a_{ij} is computed between each starting point and its K-nearest starting points including self-loop:

$$a_{ij} = \frac{exp(M(\hat{s}_i^t|\theta_u) \ominus M(s_j^{t-1}|\theta_v))}{\sum_{n=1}^{K} exp(M(\hat{s}_i^t|\theta_u) \ominus M(s_n^{t-1}|\theta_v))}, \tag{5}$$

where M denotes the MLPs with parameter θ, subscript u and v indicate two MLPs with different parameters, and $\ominus$ denotes the relation function (*i.e.*, subtraction). Finally, updated walk features are obtained by the weighted sum of the

corresponding values:

$$w_i^t = \sum_{j \in \mathcal{N}(i)} a_{ij} \odot M(v_{ij}^t | \theta_r). \quad (6)$$

Here, $\odot$ denotes Hadamard product. As a result, walk features from the last step are preserved and aggregated, the proposed Route Transformer can adaptively query the path and topology information from previous states to refine the current walk features, which enables more accurate guidance of the moving to the final destination.

3.3. Displacement Prediction

The displacement prediction module aims to use the walk features to predict a set of displacement vectors for the deformation. The common nonlocal operation [29] computes the response based on pairwise relationships. To explicitly guide the movement with current walks, we achieve this more efficiently by concatenating the inner relative features with the local feature. Given a starting point p_i with its feature x_i^t and the updated walk feature $w_i^t = \{x_i^t, x_{q,1}^t, x_{q,2}^t, ..., x_{q,l-1}^t\} \in \mathbb{R}^{l \times C}$ from Route Transformer, we define the input feature:

$$x_i^{t'} = \{[x_q^t - x_i^t, x_i^t] | q \in \mathcal{W}(i)\}, \quad (7)$$

where $[\cdot, \cdot]$ is the concatenation operation and $\mathcal{W}(i)$ indicates the other point indices in the walk w_i^t originating from the starting point x_i^t. Finally, we apply a shared MLP followed by a hyperbolic tangent activation function to produce the 3D coordinate displacement vector Δp_i^t:

$$\Delta p_i^t = tanh(MLP(x_i^{t'})). \quad (8)$$

3.4. Loss Function

In previous studies [26, 48], Chamfer Distance (CD) and Earth Mover's Distance (EMD) are the most widely used optimization loss. Initially, we exploit Chamfer Distance as our primary loss function due to its lower computation complexity which calculates the nearest distance between the entire output point cloud P' and the completed ground truth P_{gt}:

$$\mathcal{L}_{CD}(P', P_{gt}) = \sum_{x \in P'} \min_{y \in P_{gt}} \|x - y\| + \sum_{y \in P_{gt}} \min_{x \in P'} \|y - x\|. \quad (9)$$

Besides, our proposed WalkFormer only moves partial points in each deformation step, in order to match these points with the missing part, we use partial matching loss $\mathcal{L}_{PM}$ [33] which takes ground truth P_{gt} to supervise the moved starting points $P_s{}'$ in a single direction:

$$\mathcal{L}_{PM}(P_s{}', P_{gt}) = \sum_{x \in P_s{}'} \min_{y \in P_{gt}} \|x - y\|. \quad (10)$$

Therefore, the total training loss can be formulated as:

$$\mathcal{L} = \sum_t \mathcal{L}_{CD}(P^{t'}, P_{gt}) + \sum_t \mathcal{L}_{PM}(P_s^{t'}, P_{gt}), \quad (11)$$

where t denotes the deformation step.

4. Experiment

4.1. Implementation and Evaluation Metrics

Implementation. The WalkFormer adopts set abstraction and feature propagation layers [22] combined with the point transformer [50] to encode the source point cloud into point-wise features $F \in \mathbb{R}^{N \times C}$, where $N = 2048$ and $C = 128$. Then, the point cloud is down-sampled using Neighbour Similarity Sampling with $N_m = 1024$ points as the starting points for walks $W \in \mathbb{R}^{N_m \times l \times C}$, where $l = 6$ is the walk length. The number of deformation steps is set to 5, with a searching radius [35] of $\{1.0, 1.0, 0.1, 0.1, 0.01\}$.

Evaluation Metrics. In line with the existing works [26, 41, 48], we evaluate the model performance using Chamfer distance (introduced in Eq. 9). In addition, Earth Mover's Distance (EMD) [14, 27] is adopted to further evaluate the uniformity of the predicted point clouds:

$$EMD(P_1, P_2) = \min_{\phi: P_1 \to P_2} \frac{1}{|P_1|} \sum_{x \in P_1} \|x - \phi(x)\|_2, \quad (12)$$

where ϕ is a bijection.

4.2. Evaluation on Completion3D Dataset

Dataset. Completion3D [26] dataset is a widely-used point cloud completion benchmark, including 28974 training models, 800 validation models, and 1200 test models from 8 categories. Each partial point cloud is generated by back-projecting 2.5D depth images from the complete shapes into 3D. Both the partial and complete shapes consist of 2048 points. We follow the same training/validation/testing split in Completion3D for a fair comparison, where the L2 Chamfer Distance results are cited from the corresponding papers and the Earth Mover's Distance results are implemented using their open source code.

Results. We evaluate the performance of our WalkFormer against other recent point cloud completion methods, the quantitative results for each category are summarized in Table 1 and Table 2. As shown in the table, our model outperforms all counterparts in most of the categories on both CD and EMD metrics. Compared to the second-best method SeedFormer [51], WalkFormer achieves better results, reducing the average CD by 0.38 and average EMD by 0.11. As one of the few models using a point-moving strategy, our WalkFormer outperforms PMP-Net [35] and its variants PMP-Net++ [36] on all categories of this dataset,

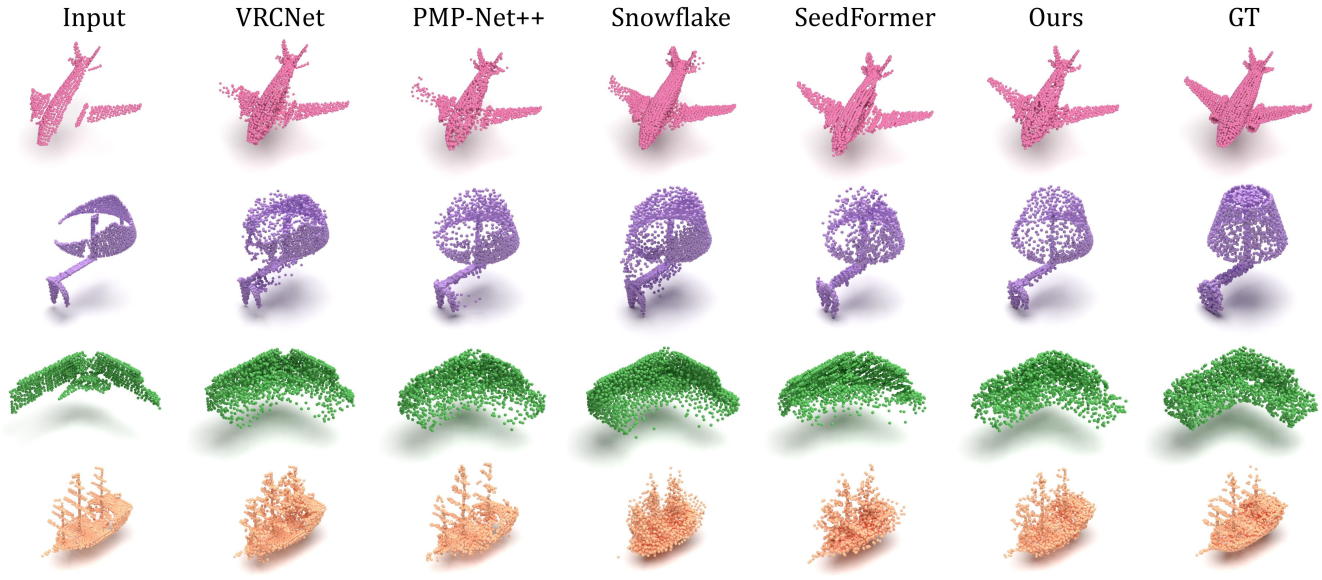

Figure 4. Visual comparison of point cloud completion results on the Completion3D dataset.

Table 1. Quantitative comparison of Completion3D dataset in terms of L2 Chamfer Distance $\times 10^4$ (lower is better).

Methods	Average	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
TopNet [26]	14.25	7.32	18.77	12.88	19.82	14.60	16.29	14.89	8.82
PMP-Net [35]	9.23	3.99	14.70	8.55	10.21	9.27	12.43	8.51	5.77
CRN [28]	9.21	3.38	13.17	8.31	10.62	10.00	12.86	9.16	5.80
VRC [19]	8.12	3.94	10.93	6.44	9.32	8.32	11.35	8.60	5.78
PMP-Net++ [36]	7.97	3.25	12.25	7.62	8.71	7.64	11.6	7.06	5.38
Snowflake [39]	7.60	3.48	11.09	6.90	8.75	8.42	10.15	**6.46**	5.32
SeedFormer [51]	6.97	2.81	10.87	**5.54**	7.90	7.18	10.46	6.75	4.32
Ours	**6.59**	**2.63**	**9.51**	6.03	**7.33**	**6.56**	**9.48**	7.01	**4.17**

Table 2. Quantitative comparison of Completion3D dataset in terms of Earth Mover's Distance $\times 10^2$ (lower is better).

Methods	Average	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
TopNet [26]	4.00	2.34	4.47	5.10	4.25	4.33	4.35	4.19	3.02
PMP-Net [35]	3.50	1.85	4.73	4.24	3.63	2.86	4.24	3.77	2.69
CRN [28]	3.46	1.96	4.43	3.15	3.91	3.87	3.64	4.14	2.58
VRC [19]	3.24	1.79	4.39	3.26	3.33	3.25	3.52	3.94	2.44
PMP-Net++ [36]	3.32	1.83	4.78	3.84	3.38	2.74	3.81	3.60	2.59
Snowflake [39]	2.99	1.57	4.20	3.23	3.08	2.77	3.44	**3.29**	2.38
SeedFormer [51]	2.86	1.55	4.28	**2.79**	2.94	2.58	3.17	3.44	2.16
Ours	**2.75**	**1.46**	**3.71**	3.02	**2.87**	**2.40**	**3.13**	3.35	**2.09**

especially regarding EMD metrics. The results demonstrate the effectiveness of our approach to producing more uniformly distributed points and motivating the points to form the missing parts.

In addition, Figure 4 visualizes the qualitative comparison results between other methods and WalkFormer. Compared with the generative-based methods like Snowflake [39] and SeedFormer [51] that decode complete point clouds from the extracted features, our deformation approach can produce more complicated topology structures as shown in the lamp and watercraft category. Other deformation-based methods like PMP-Net++ [36], take the airplane for example, points are more likely to stay around the original places. Consistent with the EMD result, our method can generate better shape completeness that outputs abundant points in the missing airplane wing.

4.3. Evaluation on PCN Dataset

Dataset. We further conduct experiments on the PCN [48] dataset. PCN dataset is derived by back-projecting ShapeNet [2] model into a 2.5D partial model from 8 viewpoints to simulate real-world incomplete data. For each shape, the complete point cloud contains 16384 points evenly sampled from the CAD model and the partial point cloud contains 2048 points as input. However, the deformation-based method requires the same resolution in both input and output point clouds. We follow [35] to solve this problem. In each step t, the partial input is concatenated with a noise vector to make the output a little different:

$$\{p_i^t\} \leftarrow \{[p_i^t, \hat{n}]\}, \hat{n} \sim \mathrm{N}(0, 1). \tag{13}$$

Here, $\mathrm{N}(0, 1)$ is a standard normal distribution. Therefore, we train our model with 2048 points and the final result consists of 8 repeated predictions for testing.

Result. Tabel 3 and Tabel 4 list the quantitative results on the PCN dataset. It shows that our proposed Walk-

Table 3. Quantitative comparison of PCN dataset in terms of L1 Chamfer Distance $\times 10^3$ (lower is better).

Methods	Average	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
TopNet [26]	12.15	7.61	13.31	10.09	13.82	14.44	14.78	11.22	11.12
PCN [48]	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
PMP-Net [35]	8.66	5.50	11.10	9.62	9.47	6.89	10.74	8.77	7.19
VRC [19]	8.17	4.78	9.96	8.52	9.14	7.42	10.82	7.24	7.49
PMP-Net++ [36]	7.56	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52
Snowflake [39]	7.21	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
SeedFormer [51]	**6.74**	3.85	**9.05**	**8.06**	**7.06**	**5.21**	8.85	**6.05**	5.85
Ours	6.79	**3.73**	9.17	8.26	7.28	5.35	**8.69**	6.12	**5.74**

Table 4. Quantitative comparison of PCN dataset in terms of Earth Mover's Distance $\times 10^2$ (lower is better).

Methods	Average	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
TopNet [26]	3.04	1.93	3.02	3.20	2.97	3.65	3.54	2.41	3.64
PCN [48]	2.99	2.36	2.98	3.17	3.19	3.78	3.06	2.11	3.34
PMP-Net [35]	2.77	1.74	2.31	3.13	3.42	3.44	3.22	1.89	3.02
VRC [19]	2.27	1.59	2.05	2.88	2.56	2.57	2.49	1.74	2.32
PMP-Net++ [36]	2.42	1.70	2.21	2.87	2.93	2.71	2.40	1.79	2.75
Snowflake [39]	2.20	1.79	**2.04**	2.67	2.40	1.98	2.86	1.86	2.01
SeedFormer [51]	2.14	**1.40**	2.70	**2.64**	2.08	**1.50**	3.19	**1.64**	1.98
Ours	**2.12**	1.64	2.26	2.93	**2.02**	2.24	**2.30**	1.75	**1.89**

Table 5. Quantitative comparison of KITTI dataset in terms of Fidelity Distance $\times 10^3$ (lower is better) and Minimal Matching Distance $\times 10^3$ (lower is better).

	PCN [48]	GRNet [41]	PoinTr [47]	SeedFormer [51]	Ours
FD	2.235	0.816	**0.000**	0.151	0.094
MMD	1.366	0.568	0.526	0.516	**0.503**

Former achieves competitive results compared to state-of-the-art methods. It is noteworthy that our model is trained on low resolution (2048 points) while testing on high resolution (16384 points), and thus this generalization ability is capable of directly applying to various resolutions. Besides, we visualize one completion process (2048 points) in Figure 5. Although our model is trained with fewer points, it is still able to recover the complete holistic shapes. During each deformation step, our method only moves a part of the points and successfully maintains the overall structure. More completion results on the PCN dataset can be found in the supplement materials.

4.4. Evaluation on KITTI Dataset

Dataset. Real-world point cloud is sparse by LiDAR scans. We follow the previous works [41, 51] to test our model on the real-scanned KITTI dataset [5]. Due to the absence of ground truth, Fidelity Distance (FD) and the Minimal Match Distance (MMD) results are used to evaluate the performance.

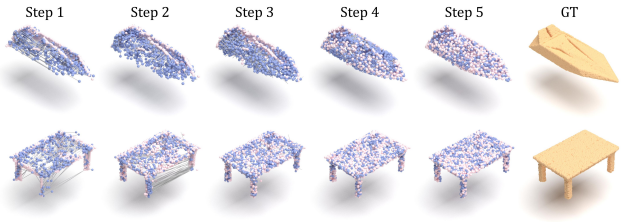

Figure 5. Visualization of the completion results in different deformation steps on PCN dataset. Blue points are moved in each step while the other points stand still.

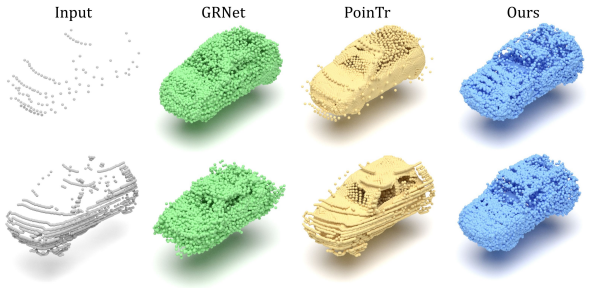

Figure 6. Visual comparison of point cloud completion results on the KITTI dataset.

Result. We fine-tune our model on ShapeNetCars and adopt the same strategy used in the PCN dataset to tackle the problem of different resolutions. The quantitative evaluations are listed in Table 5, our WalkFormer achieves the best results in MMD and has a substantial improvement in FD without merging the input. We also visualize the qualitative comparison in Figure 6, illustrating that our method is able to complete fine-grained objects on real-scanned point clouds while preserving the existing structures.

4.5. Ablation Study

In this section, we conduct a series of ablation experiments on the Completion3D dataset with 2048 points.

Neighbour Similarity Sampling. We investigate the influence of the proposed Neighbour Similarity Sampling. The quantitative results in Table 6 show a performance improvement by our sampling method. To gain further insight into the sampling process, we visualize the sampled points in Figure 7. $N_m = 128$ points are sampled for a clear comparison. Farthest point sampling (FPS) [22] is capable of selecting distant points to cover the whole set, while minimum density sampling (MDS) [14] presents an even density distribution. However, both of them tend to select the boundary points that will break the existing structures. Similar observations can be found by analysing the robustness with different numbers of sampled points N_m. When N_m is set

Table 6. Ablation study on different sampling methods.

Methods	CD-Avg	EMD-Avg
FPS	6.86	2.91
MDS	6.72	**2.73**
NSS (Ours)	**6.59**	2.75
N_m Points (NSS)	CD-Avg	EMD-Avg
$N_m = 512$	**6.55**	2.94
$N_m = 1024$	6.59	**2.75**
$N_m = 2048$	6.98	2.88

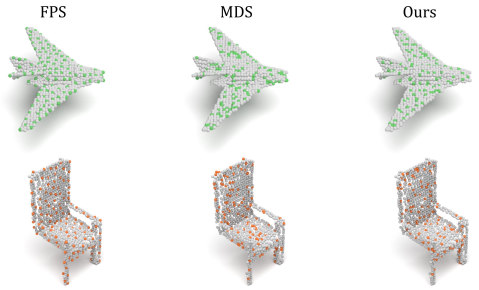

Figure 7. Qualitative comparison of the sampling results. We visualize the sampled points with a different colour.

to 2048, it means that every point is moved in each deformation step. Conversely, when N_m is reduced, the results demonstrate higher performance. This could be attributed to the fact that a subset of points is retained to preserve the holistic shape which makes the deformation process stable.

Route Transformer and Loss Function. To verify the performance of Route Transformer, we conducted experiments with five different variants, as presented in Table 7. The first variation, *NoPath*, removes the Route Transformer, thereby eliminating the skip connection between steps. Other variants, *Con* and *Add*, substitute the Route Transformer with concatenation and element-wise addition, alone with *GRU* and *RPA* from [35]. We also examine the effectiveness of Partial Matching loss ($\mathcal{L}_{\text{PM}}$) [33] and Point Moving Distance loss ($\mathcal{L}_{\text{PMD}}$) [35], while the *Baseline* is equipped with Route Transformer and solely employs the Chamfer Distance as loss function. It can be observed from the table that the features from the previous step could be useful, while simply concatenating or adding features together will lead to unsatisfactory results. Among the variations, our method demonstrates superior performance by adopting PM loss. Albeit with minor improvements to use the PMD loss, it is a strict constraint and may also limit the points to be moved over long distances.

Input Features. We study the impact of different input features in Eq. 7, including both local and long-range regions, the Route Transformer is disabled in this ablated version. For point p_i, we denote $\{f_j | j \in \mathcal{N}(i)\}$ as neighbour features from the Encoder in Sec. 3.1 and $\{f_q | q \in \mathcal{W}(i)\}$ as walk features from the Point Selector in Sec. 3.2. We set the dimension of f_j equal to f_q and remove the Point Walk module if there are no walk features as input. The experiment results are shown in Table 8, from which we can see that the original pairwise features f_i can achieve a considerable performance. Furthermore, the results demonstrate a clear improvement achieved by aggregating walk features. However, no obvious gain is obtained by neighbour features as these features have already been extracted in the Encoder layer.

Table 7. Ablation study on Route Transformer and Loss Function.

Module	CD-Avg	EMD-Avg
NoPath	6.84	2.83
Con	6.94	2.87
Add	7.05	2.92
GRU	6.79	2.85
RPA	6.73	2.79
Baseline	6.77	2.82
Baseline w/ $\mathcal{L}_{\text{PM}}$	**6.59**	**2.75**
Baseline w/ $\mathcal{L}_{\text{PMD}}$	6.72	2.82

Table 8. Performance comparisons among different input features for displacement prediction.

Input	CD-Avg	EMD-Avg
f_i	7.11	3.04
f_j, f_i	7.19	3.02
f_q, f_i	7.04	2.98
$f_j - f_i, f_i$	7.16	3.10
$f_q - f_i, f_i$	**6.84**	**2.83**
$f_j - f_i, f_q - f_i, f_i$	6.97	2.95

5. Conclusion

This paper presents WalkFormer which completes the point cloud by a multi-step partial deformation approach. Benefiting from a new sampling operation, our method can selectively move the points while preserving the existing structures. By taking guided walks on the point cloud, WalkFormer is able to model long-range interactions with topology information for predicting accurate point displacements. Extensive experiments on various datasets indicate that our model improves the point cloud completion performance compared with previous state-of-the-art methods.

Acknowledgement

This work was supported by the XJTLU Research Development Fund (RDF-21-02-080).

References

[1] Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 6

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

[4] Ben Fei, Weidong Yang, Wen-Ming Chen, Zhijun Li, Yikang Li, Tao Ma, Xing Hu, and Lipeng Ma. Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 1

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 7

[6] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1759–1769, 2020. 1, 2

[7] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021. 1, 2

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

[9] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[10] Yushi Li and George Baciu. Hsgan: Hierarchical graph learning for point cloud generation. *IEEE Transactions on Image Processing*, 30:4540–4554, 2021. 2

[11] Yushi Li and George Baciu. Sg-gan: Adversarial self-attention gcn for point cloud topological parts generation. *IEEE Transactions on Visualization and Computer Graphics*, 28(10):3499–3512, 2021. 2

[12] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018. 2

[13] Hao Liu, Yulan Guo, Yanni Ma, Yinjie Lei, and Gongjian Wen. Semantic context encoding for accurate 3d point cloud segmentation. *IEEE Transactions on Multimedia*, 23:2045–2055, 2020. 1, 2

[14] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11596–11603, 2020. 4, 5, 7

[15] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 2

[16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 4

[17] Priyanka Mandikal and Venkatesh Babu Radhakrishnan. Dense 3d point cloud reconstruction using a deep pyramid network. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1052–1060, 2019. 2

[18] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 2

[19] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8524–8533, 2021. 4, 6, 7

[20] Rolandos Alexandros Potamias, Alexandros Neofytou, Kyriaki Margarita Bintsi, and Stefanos Zafeiriou. Graphwalks: efficient shape agnostic geodesic shortest path estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2968–2977, 2022. 2, 4

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 1, 2

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 3, 4, 5, 7

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

[24] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 2

[25] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1726–1735, 2022. 1

[26] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 2, 5, 6, 7

[27] Xiaogang Wang, Marcelo H Ang, and Gim Hee Lee. Cascaded refinement network for point cloud completion with self-supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8139–8150, 2021. 2, 5

[28] 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. 2, 6

[29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2, 5

[30] 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 Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2

[31] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpoolnet: Shape descriptor for point cloud completion and classification. In *European Conference on Computer Vision*, pages 70–85, 2020. 1

[32] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Learning local displacements for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1568–1577, 2022. 1

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13080–13089, 2021. 5, 8

[34] 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, 2

[35] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7443–7452, 2021. 1, 2, 3, 5, 6, 7, 8

[36] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):852–867, 2022. 1, 2, 3, 5, 6, 7

[37] 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 *European Conference on Computer Vision*, pages 646–662, 2018. 2

[38] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2

[39] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021. 1, 2, 4, 6, 7

[40] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 915–924, 2021. 2, 4

[41] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381, 2020. 5, 7

[42] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, pages 676–693, 2022. 1, 4

[43] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. 2

[44] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2019. 4

[45] 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. 2

[46] 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. 1, 2

[47] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12498–12507, 2021. 1, 2, 7

[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, 5, 6, 7

[49] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision*, pages 512–528, 2020. 2

[50] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2, 3, 5

[51] 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 *European Conference on Computer Vision*, pages 416–432, 2022. 2, 5, 6, 7