SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer

Peng Xiang[1]*, Xin Wen[1,4]*, Yu-Shen Liu[1], Yan-Pei Cao[2], Pengfei Wan[2], Wen Zheng[2], Zhizhong Han[3]
[1]School of Software, BNRist, Tsinghua University, Beijing, China
[2]Y-tech, Kuaishou Technology, Beijing, China [3]Wayne State University [4]JD.com, Beijing, China
xp20@mails.tsinghua.edu.cn wenxin16@jd.com liuyushen@tsinghua.edu.cn
caoyanpei@gmail.com {wanpengfei,zhengwen}@kuaishou.com h312h@wayne.edu

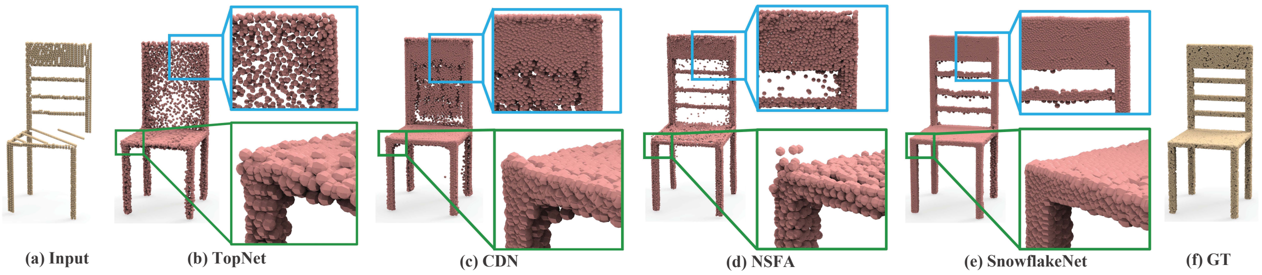


Figure 1. Visual comparison of point cloud completion results. The input and ground truth have 2048 and 16384 points, respectively. Compared with current completion methods like TopNet [43], CDN [46] and NSFA [59], our SnowflakeNet can generate the complete shape (16384 points) with fine-grained geometric details, such as smooth regions (blue boxes), sharp edges and corners (green boxes).

Abstract

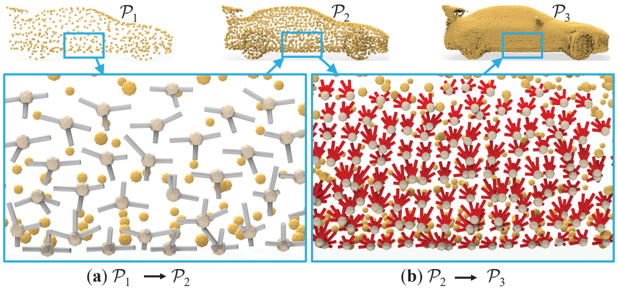
Point cloud completion aims to predict a complete shape in high accuracy from its partial observation. However, previous methods usually suffered from discrete nature of point cloud and unstructured prediction of points in local regions, which makes it hard to reveal fine local geometric details on the complete shape. To resolve this issue, we propose SnowflakeNet with Snowflake Point Deconvolution (SPD) to generate the complete point clouds. The SnowflakeNet models the generation of complete point clouds as the snowflake-like growth of points in 3D space, where the child points are progressively generated by splitting their parent points after each SPD. Our insight of revealing detailed geometry is to introduce skip-transformer in SPD to learn point splitting patterns which can fit local regions the best. Skip-transformer leverages attention mechanism to summarize the splitting patterns used in the previous SPD layer to produce the splitting in the current SPD layer. The locally compact and structured point cloud generated by SPD is able to precisely capture the structure characteristic of 3D shape in local patches, which enables the network to predict highly detailed geometries, such as smooth regions, sharp edges and corners. Our experimental results outperform the state-of-the-art point cloud completion methods under widely used benchmarks. Code will be available at `https://github.com/AllenXiangX/SnowflakeNet`.

1. Introduction

In 3D computer vision [11, 18, 20, 14, 13] applications, raw point clouds captured by 3D scanners and depth cameras are usually sparse and incomplete [53, 54, 48] due to occlusion and limited sensor resolution. Therefore, point cloud completion [53, 43], which aims to predict a complete shape from its partial observation, is vital for various downstream tasks. Benefiting from large-scaled point cloud datasets, deep learning based point cloud completion methods have been attracting more research interests. Current methods either constrain the generation of point clouds by following a hierarchical rooted tree structure [46, 55, 43] or assume a specific topology [56, 53] for the target shape. However, most of these methods suffered from discrete nature of point cloud and unstructured prediction of points in local regions, which makes it hard to preserve a well arranged structure for points in local patches. It is still challenging to capture the local geometric details and structure characteristic on the complete shape, such as smooth regions, sharp edges and corners, as illustrated in Figure 1.

In order to address this problem, we propose a novel net-

*Equal contribution. This work was supported by National Key R&D Program of China (2020YFF0304100), the National Natural Science Foundation of China (62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data. The corresponding author is Yu-Shen Liu.



$\mathcal{P}_1$
$\mathcal{P}_2$
$\mathcal{P}_3$
(a) $\mathcal{P}_1 \rightarrow \mathcal{P}_2$
(b) $\mathcal{P}_2 \rightarrow \mathcal{P}_3$

Figure 2. Illustration of snowflake point deconvolution (SPD) for growing part of a car. To show the local change more clearly, we only illustrate some sample points as parent points in the same patch and demonstrate their splitting paths for child points, which are marked as gray and red lines. (a) illustrates the SPD of point splitting from a coarse point cloud $\mathcal{P}_1$ (512 points) to its splitting $\mathcal{P}_2$ (2048 points). (b) illustrates the SPD of point splitting from $\mathcal{P}_2$ to dense complete point cloud $\mathcal{P}_3$ (16384 points), where the child points are expanding like the growth process of snowflakes.

work called *SnowflakeNet*, especially focusing on the decoding process to complete partial point clouds. SnowflakeNet mainly consists of layers of *Snowflake Point Deconvolution* (SPD), which models the generation of complete point clouds like the snowflake growth of points in 3D space. We progressively generate points by stacking one SPD layer upon another, where each SPD layer produces child points by splitting their parent point with inheriting shape characteristics captured by the parent point. Figure 2 illustrates the process of SPD and point-wise splitting.

Our insight of revealing detailed geometry is to introduce *skip-transformer* in SPD to learn point splitting patterns which can fit local regions the best. Compared with the previous methods, which often ignore the spatial relationship among points [56, 43, 31] or simply learn through self-attention in a single level of multi-step point cloud decoding [53, 29, 46], our skip-transformer is proposed to integrate the spatial relationships across different levels of decoding. Therefore, it can establish a cross-level spatial relationships between points in different decoding steps, and refine their location to produce more detailed structure. To achieve this, skip-transformer leverages attention mechanism to summarize the splitting patterns used in the previous SPD layer, which aims to produce the splitting in current SPD layer. The skip-transformer can learn the shape context and the spatial relationship between the points in local patches. This enables the network to precisely capture the structure characteristic in each local patches, and predict a better point cloud shape for both smooth plane and sharp edges in 3D space. We achieved the state-of-the-art completion accuracy under the widely used benchmarks. Our main contributions can be summarized as follows.

- We propose a novel SnowflakeNet for point cloud completion. Compared with previous locally unorganized complete shape generation methods, SnowflakeNet can interpret the generation process of complete point cloud into an explicit and locally structured pattern, which greatly improves the performance of 3D shape completion.
- We propose the novel Snowflake Point Deconvolution (SPD) for progressively increasing the number of points. It reformulates the generation of child points from parent points as a growing process of snowflake, where the shape characteristic embedded by the parent point features is extracted and inherited into the child points through a *point-wise splitting* operation.
- We introduce a novel skip-transformer to learn splitting patterns in SPD. It learns shape context and spatial relationship between child points and parent points, which encourages SPD to produce locally structured and compact point arrangements, and capture the structure characteristic of 3D surface in local patches.

2. Related Work

Point cloud completion methods can be roughly divided into two categories. (1) Traditional point cloud completion methods [42, 1, 44, 26] usually assume a smooth surface of 3D shape, or utilize large-scaled complete shape dataset to infer the missing regions for incomplete shape. (2) Deep learning [28, 12, 15, 23, 19, 17, 16, 22, 50, 51] based methods [27, 46, 4, 8, 37, 25, 24], however, learn to predict a complete shape based on the learned prior from the training data. Our method falls into the second class and focuses on the decoding process of point cloud completion. We briefly review deep learning based methods below.

Point cloud completion by folding-based decoding. The development of deep learning based 3D point cloud processing techniques [10, 52, 49, 35, 33, 34, 32, 21, 3, 36] have boosted the research of point cloud completion. Suffering from the discrete nature of point cloud data, the generation of high-quality complete shape is one of the major concerns in the point cloud completion research. One of the pioneering work is the FoldingNet [56], although it was not originally designed for point cloud completion. It proposed a two-stage generation process and combined with the assumption that 3D object lies on 2D-manifold [43]. Following the similar practice, methods like SA-Net [53] further extended such generation process into multiple stages by proposing hierarchical folding in decoder. However, the problem of these folding-based methods [53, 56, 30] is that the 3-dimensional code generated by intermediate layer of network is an implicit representation of target shape, which is hardly interpreted or constrained in order to help refine the shape in local region. On the other hand, TopNet [43] modeled the point cloud generation process as the growth of rooted tree, where one parent point feature is projected into several child point features in a feature expansion layer of TopNet. Same as FoldingNet [56], the intermediate generation processes of TopNet and SA-Net are also implicit,

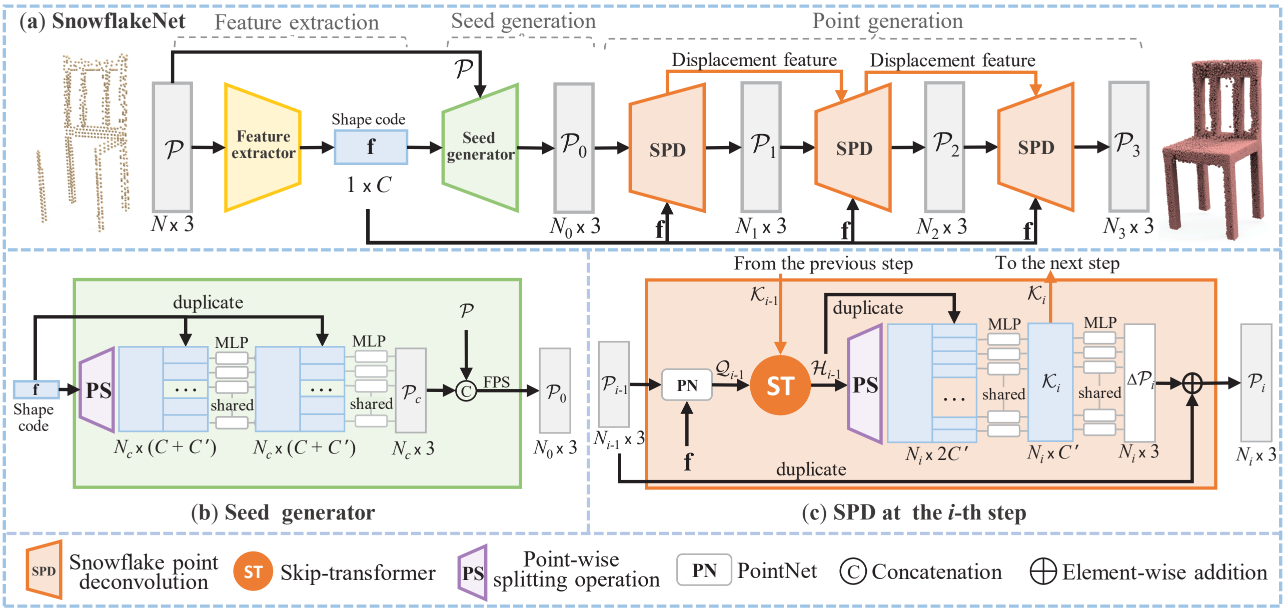


Figure 3. (a) The overall architecture of SnowflakeNet, which consists of three modules: feature extraction, seed generation and point generation. (b) The details of seed generation module. (c) Snowflake point deconvolution (SPD). Note that N, N_c and N_i are the number of points, C and C' are the number of point feature channels that are 512 and 128, respectively.

where the shape information is only represented by the point features, cannot be constrained or explained explicitly.

Point cloud completion by coarse-to-fine decoding. Recently, explicit coarse-to-fine completion framework [55, 5] has received an increasing attention, due to its explainable nature and controllable generation process. Typical methods like PCN [57] and NSFA [59] adopted the two-stage generation framework, where a coarse and low resolution point cloud is first generated by the decoder, and then a lifting module is used to increase the density of point clouds. Such kind of methods can achieve better performance since it can impose more constraints on the generation process of point cloud, i.e. the coarse one and the dense one. Followers like CDN [46] and PF-Net [27] further extended the number of generation stages and achieved the currently state-of-the-art performance. Although intriguing performance has been achieved by the studies along this line, most of these methods still cannot predict a locally structure point splitting pattern, as illustrated in Figure 1. The biggest problem is that these methods only focus on the expansion of point number and the reconstruction of global shape, while ignoring to preserve a well-structured generation process for points in local regions. This makes these methods difficult to capture local detailed geometries and structures of 3D shape.

Compared with the above-mentioned methods, our SnowflakeNet takes one step further to explore an explicit, explainable and locally structured solution for the generation of complete point cloud. SnowflakeNet models the progressive generation of point cloud as a hierarchical rooted tree structure like TopNet, while keeping the process explainable and explicit like CDN [46] and PF-Net [27]. Moreover, it excels the predecessors by arranging the point splitting in local regions in a locally structured pattern, which enables to precisely capture the detailed geometries and structures of 3D shapes.

Relation to transformer. Transformer [45] was initially proposed for encoding sentence in natural language processing, and soon gets popular in the research of 2D computer vision (CV) [6, 39]. Then, the success of transformer-based 2D CV studies have drawn the attention of 3D point cloud research, where pioneering studies like Point-Transformer [60], PCT [9] and Pointformer [38] have introduced such framework in the encoding process of point cloud to learn the representation. In our work, instead of only utilizing its representation learning ability, we further extend the application of transformer-based structure into the decoding process of point cloud completion, and reveal its ability for generating high quality 3D shapes through the proposed skip-transformer.

3. SnowflakeNet

The overall architecture of SnowflakeNet is shown in Figure 3(a), which consists of three modules: feature extraction, seed generation and point generation. We will detail each module in the following.

3.1. Overview

Feature extraction module. Let $\mathcal{P} = \{\mathbf{p}_j\}$ of size $N \times 3$ be an input point cloud, where N is the number of points and each point $\mathbf{p}_j$ indicates a 3D coordinate. The feature extractor aims to extract a shape code $\mathbf{f}$ of size $1 \times C$, which captures the global structure and detailed local pattern of the target shape. To achieve this, we adopt three layers of set abstraction from [41] to aggregate point features from local to global, along which point transformer [60] is applied to incorporate local shape context.

Seed generation module. The objective of the seed generator is to produce a coarse but complete point cloud $\mathcal{P}_0$ of size $N_0 \times 3$ that captures the geometry and structure of the target shape. As shown in Figure 3(b), with the extracted shape code $\mathbf{f}$, the seed generator first produces point features that capture both the existing and missing shape through point-wise splitting operation. Next, the per-point features are integrated with the shape code through multi-layer perceptron (MLP) to generate a coarse point cloud $\mathcal{P}_c$ of size $N_c \times 3$. Then, following the previous method [46], $\mathcal{P}_c$ is merged with the input point cloud $\mathcal{P}$ by concatenation, and then the merged point cloud is down-sampled to $\mathcal{P}_0$ through farthest point sampling (FPS) [41]. In this paper, we typically set $N_c = 256$ and $N_0 = 512$, where a sparse point cloud $\mathcal{P}_0$ suffices for representing the underlying shape. $\mathcal{P}_0$ will serve as the seed point cloud for point generation module.

Point generation module. The point generation module consists of three steps of Snowflake Point Deconvolution (SPD), each of which takes the point cloud from the previous step and splits it by up-sampling factors (denoted by r_1, r_2 and r_3) to obtain $\mathcal{P}_1, \mathcal{P}_2$ and $\mathcal{P}_3$, which have the point sizes of $N_1 \times 3$, $N_2 \times 3$ and $N_3 \times 3$, respectively. SPDs collaborate with each other to generate a rooted tree structure that complies with local pattern for every seed point. The structure of SPD is detailed below.

3.2. Snowflake Point Deconvolution (SPD)

The SPD aims to increase the number of points by splitting each parent point into multiple child points, which can be achieved by first duplicating the parent points and then adding variations. Existing methods [46, 57, 59] usually adopt the folding-based strategy [56] to obtain the variations, which are used for learning different displacements for the duplicated points. However, the folding operation samples the same 2D grids for each parent point, which ignores the local shape characteristics contained in the parent point. Different from the folding-based methods [46, 57, 59], the SPD obtains variations through a *point-wise splitting* operation, which fully leverages the geometric information in parent points and adds variations that comply with local patterns. In order to progressively generate the split points, three SPDs are used in point generation module. In addition, to facilitate consecutive SPDs to split points in a coherent manner, we propose a novel skip-transformer to capture the shape context and the spatial relationship between the parent points and their split points.

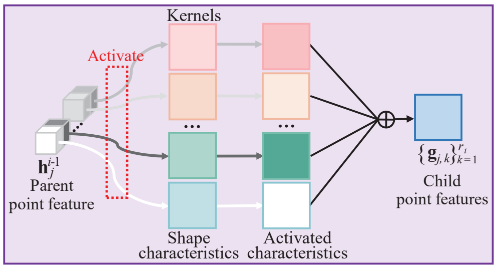


Figure 4. The point-wise splitting operation. The cubes are logits of the parent point feature that represent activation status of the corresponding shape characteristics (Kernels), and child point features are obtained by adding activated shape characteristics.

Figure 3(c) illustrates the structure of the i-th SPD with up-sampling factor r_i. We denote a set of parent points obtained from previous step as $\mathcal{P}_{i-1} = \{\mathbf{p}_j^{i-1}\}_{j=1}^{N_{i-1}}$. We split the parent points in $\mathcal{P}_{i-1}$ by duplicating them r_i times to generate a set of child points $\hat{\mathcal{P}}_i$, and then spread $\hat{\mathcal{P}}_i$ to the neighborhood of the parent points. To achieve this, we take the inspiration from [57] to predict the *point displacement* $\Delta\mathcal{P}_i$ of $\hat{\mathcal{P}}_i$. Then, $\hat{\mathcal{P}}_i$ is updated as $\mathcal{P}_i = \hat{\mathcal{P}}_i + \Delta\mathcal{P}_i$, where $\mathcal{P}_i$ is the output of the i-th SPD.

In detail, taking the shape code $\mathbf{f}$ from feature extraction, the SPD first extracts the per-point feature $\mathcal{Q}_{i-1} = \{\mathbf{q}_j^{i-1}\}_{j=1}^{N_{i-1}}$ for $\mathcal{P}_{i-1}$ by adopting the basic PointNet [40] framework. Then, $\mathcal{Q}_{i-1}$ is sent to the skip-transformer to learn the *shape context feature*, denoted as $\mathcal{H}_{i-1} = \{\mathbf{h}_j^{i-1}\}_{j=1}^{N_{i-1}}$. Next, $\mathcal{H}_{i-1}$ is up-sampled by *point-wise splitting* operator and duplication, respectively, where the former serves to add variations and the latter preserves shape context information. Finally, the up-sampled feature with size of $N_i \times 2C'$ is fed to MLP to produce the *displacement feature* $\mathcal{K}_i = \{\mathbf{k}_j^i\}_{j=1}^{N_i}$ of current step. Here, $\mathcal{K}_i$ is used for generating the point displacement $\Delta\mathcal{P}_i$, and will be fed into the next SPD. $\Delta\mathcal{P}_i$ is formulated as

$$\Delta\mathcal{P}_i = \tanh(\mathrm{MLP}(\mathcal{K}_i)), \tag{1}$$

where $\tanh$ is the hyper-tangent activation.

Point-wise splitting operation. point-wise splitting operation aims to generate multiple child point features for each $\mathbf{h}_j^{i-1} \in \mathcal{H}_{i-1}$, $j = 1, N_{i-1}$. Figure 4 shows this operation structure used in i-th SPD (see Figure 3(c)). It is a special one-dimensional deconvolution strategy, where the kernel size and stride are both equal to r_i. In practice, each $\mathbf{h}_j^{i-1} \in \mathcal{H}_{i-1}$shares the same set of kernels, and produces multiple child point features in a point-wise manner. To be clear, we denote the m-th logit of $\mathbf{h}_j^{i-1}$ as $h_{j,m}^{i-1}$, and its corresponding kernel is indicated by K_m. Technically, K_m is a matrix with a size of $r_i \times \mathrm{C}'$, the k-th row of K_m is denoted as $\mathbf{k}_{m,k}$, and the k-th child point feature $\mathbf{g}_{j,k}$ is given by

$$\mathbf{g}_{j,k} = \sum_m h_{j,m}^{i-1}\mathbf{k}_{m,k}. \tag{2}$$

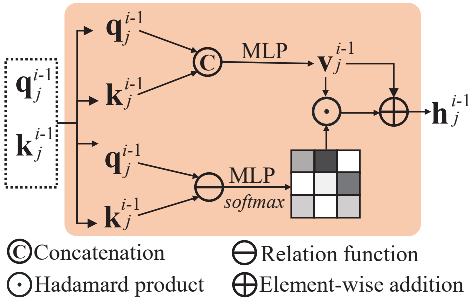


Figure 5. The detailed structure of skip-transformer.

In addition, in Figure 4, we assume that each learnable kernel K_m indicates a certain shape characteristic, which describes the geometry and structure of 3D shape in local region. Correspondingly, every logit $h^{i-1}_{j,m}$ indicates the activation status of the m-th shape characteristic. The child point features can be generated by adding the activated shape characteristics. Moreover, the point-wise splitting operation is flexible for up-sampling points. For example, when $r_i = 1$, it enables the SPD to move the point from previous step to a better position; when $r_i > 1$, it serves to expand the number of points by a factor of r_i.

Collaboration between SPDs. In Figure 3(a), we adopt three SPDs to generate the complete point cloud. We first set the up-sampling factor $r_1 = 1$ to explicitly rearrange seed point positions. Then, we set $r_2 > 1$ and $r_3 > 1$ to generate a structured tree for every point in $\mathcal{P}_1$. Collaboration between SPDs is crucial for growing the tree in a coherent manner, because information from the previous splitting can be used to guide the current one. Besides, the growth of the rooted trees should also capture the pattern of local patches to avoid overlapping with each other. To achieve this purpose, we propose a novel *skip-transformer* to serve as the cooperation unit between SPDs. In Figure 5, the skip-transformer takes per-point feature $\mathbf{q}^{i-1}_j$ as input, and combines it with displacement feature $\mathbf{k}^{i-1}_j$ from previous step to produce the shape context feature $\mathbf{h}^{i-1}_j$, which is given by

$$\mathbf{h}^{i-1}_j = \mathrm{ST}(\mathbf{k}^{i-1}_j, \mathbf{q}^{i-1}_j), \tag{3}$$

where ST denotes the skip-transformer. The detailed structure is described as follows.

3.3. Skip-Transformer

Figure 5 shows the structure of skip-transformer. The skip-transformer is introduced to learn and refine the spatial context between parent points and their child points, where the term “skip” represents the connection between the displacement feature from the previous layer and the point feature of the current layer.

Given per-point feature $\mathbf{q}^{i-1}_j$ and displacement feature $\mathbf{k}^{i-1}_j$, the skip-transformer first concatenates them. Then, the concatenated feature is fed to MLP, which generates the vector $\mathbf{v}^{i-1}_j$. Here, $\mathbf{v}^{i-1}_j$ serves as the value vector which incorporates previous point splitting information. In order to further aggregate local shape context into $\mathbf{v}^{i-1}_j$, the skip-transformer uses $\mathbf{q}^{i-1}_j$ as the query and $\mathbf{k}^{i-1}_j$ as the key to estimate attention vector $\mathbf{a}^{i-1}_j$, where $\mathbf{a}^{i-1}_j$ denotes how much attention the current splitting should pay to the previous one. To enable the skip-transformer to concentrate on local pattern, we calculate attention vectors between each point and its k-nearest neighbors (k-NN). The k-NN strategy also helps to reduce computation cost. Specifically, given the j-th point feature $\mathbf{q}^{i-1}_j$, the attention vector $\mathbf{a}^{i-1}_{j,l}$ between $\mathbf{q}^{i-1}_j$ and displacement features of the k-nearest neighbors $\{\mathbf{k}^{i-1}_{j,l} | l = 1, 2, \dots, k\}$ can be calculated as

$$\mathbf{a}^{i-1}_{j,l} = \frac{\exp(\mathrm{MLP}((\mathbf{q}^{i-1}_j) \ominus (\mathbf{k}^{i-1}_{j,l})))}{\sum_{l=1}^{k} \exp(\mathrm{MLP}((\mathbf{q}^{i-1}_j) \ominus (\mathbf{k}^{i-1}_{j,l})))}, \tag{4}$$

where $\ominus$ serves as the relation operation, i.e. element-wise subtraction. Finally, the shape context feature $\mathbf{h}^{i-1}_j$ can be obtained by

$$\mathbf{h}^{i-1}_j = \mathbf{v}^{i-1}_j \oplus \sum_{l=1}^{k} \mathbf{a}^{i-1}_{j,l} \odot \mathbf{v}^{i-1}_{j,l}, \tag{5}$$

where $\oplus$ denotes element-wise addition and $\odot$ is Hadamard product. Note that there is no previous displacement feature for the first SPD, of which the skip-transformer takes $\mathbf{q}^0_j$ as both query and key.

3.4. Training Loss

In our implementation, we use Chamfer distance (CD) as the primary loss function. To explicitly constrain point clouds generated in the seed generation and the subsequent splitting process, we down-sample the ground truth point clouds to the same sampling density as $\{\mathcal{P}_c, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$ (see Figure 3), where we define the sum of the four CD losses as the *completion loss*, denoted by $\mathcal{L}_{\text{completion}}$. Besides, we also exploit the *partial matching loss* from [48] to preserve the shape structure of the input point cloud. It is an unidirectional constraint which aims to match one shape to the other without constraining the opposite direction. Because the partial matching loss only requires the output point cloud to partially match the input, we take it as the *preservation loss* $\mathcal{L}_{\text{preservation}}$, and the total training loss is formulated as

$$\mathcal{L} = \mathcal{L}_{\text{completion}} + \lambda \mathcal{L}_{\text{preservation}}. \tag{6}$$

The arrangement is detailed in *Supplementary Material*.

4. Experiments

To fully prove the effectiveness of our SnowflakeNet, we conduct comprehensive experiments under two widely used benchmarks:PCN [57] and Completion3D [43], both

Table 1. Point cloud completion on PCN dataset in terms of per-point L1 Chamfer distance $\times 10^3$ (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat
FoldingNet [56]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet [43]	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet [7]	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
PCN [57]	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
GRNet [55]	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
CDN [46]	8.51	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05
PMP-Net [54]	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
NSFA [59]	8.06	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48
Ours	**7.21**	**4.29**	**9.16**	**8.08**	**7.89**	**6.07**	**9.23**	**6.55**	**6.40**

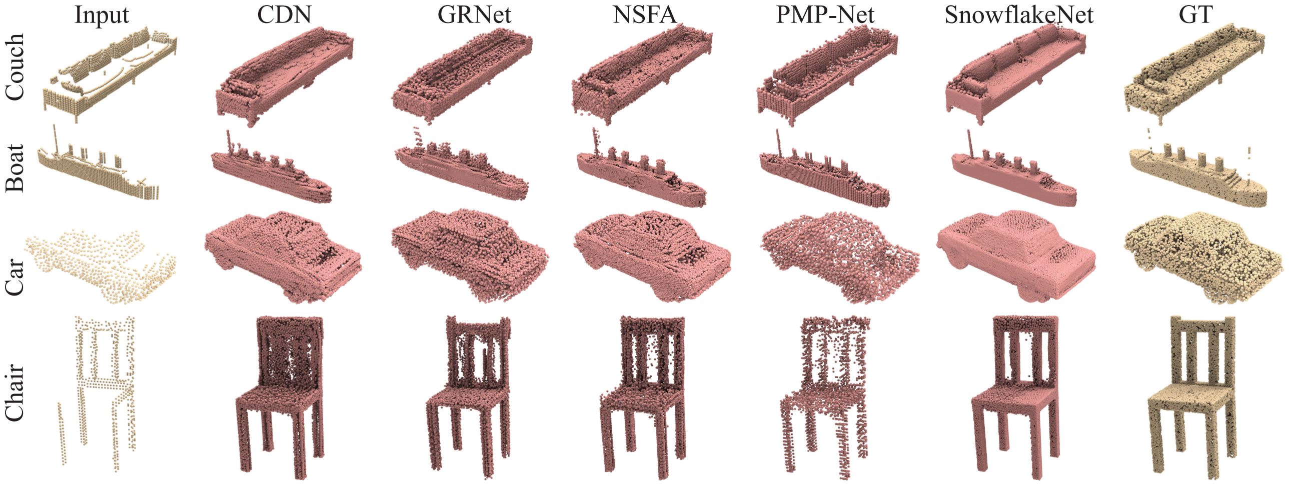


Figure 6. Visual comparison of point cloud completion on PCN dataset. Our SnowflakeNet can produce smoother surfaces (e.g. car) and more detailed structures (e.g. chair back) compared with the other state-of-the-art point cloud completion methods.

of which are subsets of the ShapeNet dataset. The experiments demonstrate that our method has superiority over the state-of-the-art point cloud completion methods.

4.1. Evaluation on PCN Dataset

Dataset briefs and evaluation metric. The *PCN* dataset [57] is a subset with 8 categories derived from ShapeNet dataset [2]. The incomplete shapes are generated by back-projecting complete shapes into 8 different partial views. For each complete shape, 16384 points are evenly sampled from the shape surface. We follow the same split settings with PCN [57] to fairly compare our SnowflakeNet with other methods. For evaluation, we adopt the L1 version of Chamfer distance, which follows the same practice as previous methods [57].

Quantitative comparison. Table 1 shows the results of our SnowflakeNet and other completion methods on PCN dataset, from which we can find that SnowflakeNet achieves the best performance over all counterparts. Especially, compared with the result of the second-ranked NSFA [59], SnowflakeNet reduces the average CD by 0.85, which is 10.5% lower than the NSFA's results (8.06 in terms of average CD). Moreover, SnowflakeNet also achieves the best results on all categories in terms of CD, which proves the robust generalization ability of SnowflakeNet for completing shapes across different categories. In Table 1, both CDN [46] and NSFA [59] are typical point cloud completion methods, which adopt a coarse-to-fine shape decoding strategy and model the generation of points as a hierarchical rooted tree. Compared with these two methods, our SnowflakeNet also adopts the same decoding strategy but achieves much better results on PCN dataset. Therefore, the improvements should credit to the proposed SPD layers and skip-transformer in SnowflakeNet, which helps to generate points in local regions in a locally structured pattern.

Visual comparison. We typically choose top four point cloud completion methods from Table 1, and visually compare SnowflakeNet with these methods in Figure 6. The visual results show that SnowflakeNet can predict the complete point clouds with much better shape quality. For example, in the car category, the point distribution on the car's boundary generated by SnowflakeNet is smoother and more uniform than other methods. As for the chair category, SnowflakeNet can predict more detailed and clear structure

Table 2. Point cloud completion on Completion3D in terms of per-point L2 Chamfer distance $\times 10^4$ (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat
FoldingNet [56]	19.07	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51
PCN [57]	18.22	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73
PointSetVoting [58]	18.18	6.88	21.18	15.78	22.54	18.78	28.39	19.96	11.16
AtlasNet [7]	17.77	10.36	23.40	13.40	24.16	20.24	20.82	17.52	11.62
SoftPoolNet [47]	16.15	5.81	24.53	11.35	23.63	18.54	20.34	16.89	7.14
TopNet [43]	14.25	7.32	18.77	12.88	19.82	14.60	16.29	14.89	8.82
SA-Net [53]	11.22	5.27	14.45	7.78	13.67	13.53	14.22	11.75	8.84
GRNet [55]	10.64	6.13	16.90	8.27	12.23	10.22	14.93	10.08	5.86
PMP-Net [54]	9.23	3.99	14.70	8.55	10.21	9.27	12.43	8.51	5.77
Ours	**7.60**	**3.48**	**11.09**	**6.9**	**8.75**	**8.42**	**10.15**	**6.46**	**5.32**

of the chair back compared with the other methods, where CDN [46] almost fails to preserve the basic structure of the chair back, while the other methods generate lots of noise between the columns of the chair back.

4.2. Evaluation on Completion3D Dataset

Dataset briefs and evaluation metric. The Completion3D dataset contains 30958 models from 8 categories, of which both partial and ground truth point clouds have 2048 points. We follow the same train/validation/test split of Completion3D to have a fair comparison with the other methods, where the training set contains 28974 models, validation and testing set contain 800 and 1184 models, respectively. For evaluation, we adopt the L2 version of Chamfer distance on testing set to align with previous studies.

Quantitative comparison. In Table 2, we show the quantitative results of our SnowflakeNet and the other methods on Completion3D dataset. All results are cited from the online public leaderboard of Completion3D[*]. From Table 2, we can find that our SnowflakeNet achieves the best results over all methods listed on the leaderboard. Especially, compared with the state-of-the-art method PMP-Net [54], SnowflakeNet significatly reduces the average CD by 1.63, which is 17.3% lower than the PMP-Net (9.23 in terms of average CD). On the Completion3D dataset, SnowflakeNet outperforms the other methods in all categories in terms of per-category CD. Especially in the cabinet category, SnowflakeNet reduces the per-category CD by 3.61 compared with the second-ranked result of PMP-Net. Compared with PCN dataset, the point cloud in Completion3D dataset is much sparser and easier to generate. Therefore, a coarse-to-fine decoding strategy may have less advantages over the other methods. Despite of this, our SnowflakeNet still achieves the superior performance over folding-based methods including SA-Net [53] and FoldingNet [56], and we are also the best among the coarse-to-fine methods including TopNet [43] and GRNet [55]. In all, the results on Completion3D dataset demonstrate the superior capability of SnowflakeNet for predicting high-quality complete shape on sparse point clouds.

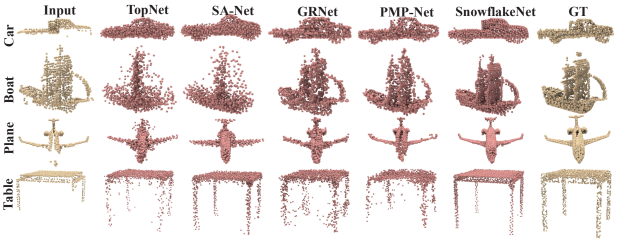


Figure 7. Visual comparison of point cloud completion on Completion3D dataset. Our SnowflakeNet can produce smoother surfaces (e.g. car and table) and more detailed structures compared with the other state-of-the-art point cloud completion methods.

Visual comparison. Same as the practice in PCN dataset, we also visually compare SnowflakeNet with the top four methods in Table 2. Visual comparison in Figure 7 demonstrates that our SnowflakeNet also achieves much better visual results than the other counterparts on sparse point cloud completion task. Especially, in plane category, SnowflakeNet predicts the complete plane which is almost the same as the ground truth, while the other methods fail to reveal the complete plane in detail. The same conclusion can also be drawn from the observation of car category. In the table and boat categories, SnowflakeNet produces more detailed structures compared with the other methods, e.g. the sails of the boat and the legs of the table.

4.3. Ablation studies

We analyze the effectiveness of each part of SnowflakeNet. For convenience, we conduct all experiments on the validation set of Completion3D dataset. By default, all the experiment settings and the network structure remains the same as Section 4.2, except for the analyzed part.

Effect of skip-transformer. To evaluate the effectiveness of skip-transformer used in SnowflakeNet, we develop three network variations as follows. (1) The *Self-att* variation replaces the transformer mechanism in skip-transformer with the self-attention mechanism, where the input is the point features of current layer. (2) The *No-att* variation removes

[*] https://completion3d.stanford.edu/results

the transformer mechanism from skip-transformer, where the features from the previous layer of SPD is directly added to the feature of current SPD layer. (3) The *No-connect* variation removes the whole skip-transformer from the SPD layers, and thus, no feature connection is established between the SPD layers. The experiment results are shown in Table 3. In addition, we denote the original version of SnowflakeNet as *Full* for clear comparison with the performance of each network variations. From Table 3, we can find that the transformer-based Full model achieves the best performance among all compare network variations. The comparison between the No-connect model and the Full model justifies the advantage of using skip-transformer between SPD layers, and the comparison between No-att model and Full model further proves the effectiveness of using transformer mechanism to learn shape context in local regions. Moreover, the comparison between Self-att model and No-att model shows that the attention based mechanism can also contribute to the completion performance.

Table 3. Effect of skip-transformer.

Methods	avg.	Couch	Chair	Car	Lamp
Self-att	8.89	6.04	10.9	9.42	9.12
No-att	9.30	6.15	11.2	10.4	9.38
No-connect	9.39	6.17	11.3	10.5	9.51
Full	**8.48**	**5.89**	**10.6**	**9.32**	**8.12**

Effect of each part in SnowflakeNet. To evaluate the effectiveness of each part in SnowflakeNet, we design four different network variations as follows. (1) The *Folding-expansion* variation replaces the point-wise splitting operation with the folding-based feature expansion method [56], where the features are duplicated several times and concatenated with a 2-dimensional codeword, in order to increase the number of point features. (2) The $\mathrm{E_{PCN}}$+*SPD* variation employs the PCN encoder and our SPD with skip-transformer as decoder. (3) The *w/o partial matching* variation removes the partial matching loss. (4) The *PCN-baseline* is the performance of original PCN method [57], which is trained and evaluated under the same settings of our ablation study. In Table 4, we report the results of the four network variations along with the default network denoted as *Full*. By comparing $\mathrm{E_{PCN}}$+SPD with PCN-baseline, we can find that our SPD with skip-transformer based decoder can be potentially applied to other simple encoders, and achieves significant improvement. By comparing Folding-expansion with Full model, the better performance of Full model proves the advantage of point-wise splitting operation over the folding-based feature expansion methods. By comparing w/o partial matching with Full model, we can find that partial matching loss can slightly improve the average performance of SnowflakeNet.

Visualization of point generation process of SPD. In Figure 8, we visualize the point cloud generation proccess of SPD. We can find that the layers of SPD generate points in a snowflake-like pattern. When generating the smooth plane (e.g. chair and lamp in Figure 8), we can clearly see the child points are generated around the parent points, and smoothly placed along the plane surface. On the other hand, when generating thin tubes and sharp edges, the child points can precisely capture the geometries.

Table 4. Effect of each part in SnowflakeNet.

Methods	avg.	Couch	Chair	Car	Lamp
Folding-expansion	8.80	8.40	10.80	5.83	10.10
$\mathrm{E_{PCN}}$+SPD	8.93	9.06	11.30	6.14	9.23
w/o partial matching	8.50	8.72	**10.6**	**5.78**	**8.9**
PCN-baseline	13.30	11.50	17.00	6.55	18.20
Full	**8.48**	**8.12**	10.6	5.89	9.32

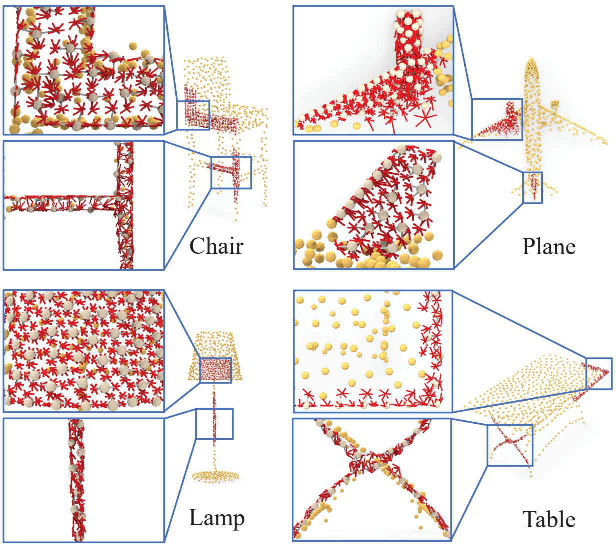


Figure 8. Visualization of snowflake point deconvolution on different objects. For each object, we sample two patches of points and visualize two layers of point splitting together for each sampled point. The gray lines indicate the paths of the point splitting from $\mathcal{P}_1$ to $\mathcal{P}_2$, and the red lines are splitting paths from $\mathcal{P}_2$ to $\mathcal{P}_3$.

5. Conclusions

In this paper, we propose a novel neural network for point cloud completion, named SnowflakeNet. The SnowflakeNet models the generation of completion point clouds as the snowflake-like growth of points in 3D space using multiple layers of Snowflake Point Deconvolution. By further introducing skip-transformer in Snowflake Point Deconvolution, SnowflakeNet learns to generate locally compact and structured point cloud with highly detailed geometries. We conduct comprehensive experiments on sparse (Completion3D) and dense (PCN) point cloud completion datasets, which shows the superiority of our SnowflakeNet over the current SOTA point cloud completion methods.

References

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

[2] Angel X Chang, Thomas Funkhouser, Leonidas J 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:1512.03012*, 2015. 6

[3] Chao Chen, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Unsupervised learning of fine structure generation for 3D point clouds by 2D projection matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[4] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. In *International Conference on Learning Representations*, 2019. 2

[5] 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 (CVPR)*, pages 5868–5877, 2017. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[7] Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018. 6, 7

[8] Jiayuan Gu, Wei-Chiu Ma, Sivabalan Manivasagam, Wenyuan Zeng, Zihao Wang, Yuwen Xiong, Hao Su, and Raquel Urtasun. Weakly-supervised 3D shape completion in the wild. In *Proc. of the European Conf. on Computer Vision (ECCV)*. Springer, 2020. 2

[9] 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(2):187–199, Apr 2021. 3

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

[11] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1018–1027, 2020. 1

[12] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *International Joint Conference on Artificial Intelligence*, 2019. 2

[13] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation. *IEEE Transactions on Image Processing*, 26(8):3707–3720, 2017. 1

[14] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. Deep Spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. *IEEE Transactions on Image Processing*, 27(6):3049–3063, 2018. 1

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

[16] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. Reconstructing 3D shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 29:8721–8734, 2020. 2

[17] Zhizhong Han, Guanhui Qiao, Yu-Shen Liu, and Matthias Zwicker. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision*, pages 607–625. Springer, 2020. 2

[18] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *The 33rd AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 1

[19] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by RNN with attention. *IEEE Transactions on Image Processing*, 28(2):658–672, 2018. 2

[20] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. *The 33th AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 1

[21] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-angle point cloud-vae: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10441–10450. IEEE, 2019. 2

[22] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Hierarchical view predictor: Unsupervised 3D global feature learning through hierarchical prediction among unordered views. In *ACM International Conference on Multimedia*, 2021. 2

[23] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and CL Chen. 3DViewGraph: Learning global features for 3D shapes from a graph of unordered views with attention. In *International Joint Conference on Artificial Intelligence*, 2019. 2

[24] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 2

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

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

[27] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7662–7670, 2020. 2, 3

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

[29] Yu Lequan, Li Xianzhi, Fu Chi-Wing, Cohen-Or Daniel, and Heng Pheng-Ann. PU-Net: Point cloud upsampling network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2018. 2

[30] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-GAN: A point cloud upsampling adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 7203–7212, 2019. 2

[31] 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*, volume 34, pages 11596–11603, 2020. 2

[32] Xinhai Liu, Zhizhong Han, Fangzhou Hong, Yu-Shen Liu, and Matthias Zwicker. LRC-Net: Learning discriminative features on point clouds by encoding local region contexts. *Computer Aided Geometric Design*, 79:101859, 2020. 2

[33] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8778–8785, 2019. 2

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

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

[36] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces. In *International Conference on Machine Learning (ICML)*, 2021. 2

[37] Yinyu Nie, Yiqun Lin, Xiaoguang Han, Shihui Guo, Jian Chang, Shuguang Cui, and Jian.J Zhang. Skeleton-bridged point completion: From global inference to local adjustment. In *Advances in Neural Information Processing Systems*, volume 33, pages 16119–16130. Curran Associates, Inc., 2020. 2

[38] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3D object detection with Pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7463–7472, June 2021. 3

[39] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 3

[40] 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 (CVPR)*, pages 1063–6919, 2017. 4

[41] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 4

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

[43] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. TopNet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 383–392, 2019. 1, 2, 5, 6, 7

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

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. 3

[46] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEEF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 790–799, 2020. 1, 2, 3, 4, 6, 7

[47] 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 (ECCV)*, 2020. 7

[48] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 5

[49] Xin Wen, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu. Point2SpatialCapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. *IEEE Transactions on Image Processing*, 29:8855–8869, 2020. 2

[50] Xin Wen, Zhizhong Han, and Yu-Shen Liu. CMPD: Using cross memory network with pair discrimination for image-text retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(6):2427–2437, 2020. 2

[51] Xin Wen, Zhizhong Han, Xinyu Yin, and Yu-Shen Liu. Adversarial cross-modal retrieval via learning and transferring single-modal similarities. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 478–483. IEEE, 2019. 2

[52] Xin Wen, Zhizhong Han, Geunhyuk Youk, and Yu-Shen Liu. CF-SIS: Semantic-instance segmentation of 3D point clouds by context fusion with self-attention. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1661–1669, 2020. 2

[53] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1939–1948, 2020. 1, 2, 7

[54] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 6, 7

[55] 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 (ECCV)*, 2020. 1, 3, 6, 7

[56] 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 (CVPR)*, pages 206–215, 2018. 1, 2, 4, 6, 7, 8

[57] 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. 3, 4, 5, 6, 7, 8

[58] Junming Zhang, Weijia Chen, Yuping Wang, Ram Vasudevan, and Matthew Johnson-Roberson. Point set voting for partial point cloud analysis. *IEEE Robotics and Automation Letters*, 2021. 7

[59] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision (ECCV)*, pages 512–528, 2020. 1, 3, 4, 6

[60] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 3, 4