

LAKe-Net: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints

Junshu Tang¹, Zhijun Gong¹, Ran Yi^{1*}, Yuan Xie², Lizhuang Ma^{1,2*}

¹Shanghai Jiao Tong University, ²East China Normal University

{tangjs, gongzhijun, ranyi}@sjtu.edu.cn, yxie@cs.ecnu.edu.cn, ma-lz@cs.sjtu.edu.cn

Abstract

Point cloud completion aims at completing geometric and topological shapes from a partial observation. However, some topology of the original shape is missing, existing methods directly predict the location of complete points, without predicting structured and topological information of the complete shape, which leads to inferior performance. To better tackle the missing topology part, we propose LAKe-Net, a novel topology-aware point cloud completion model by localizing aligned keypoints, with a novel **Keypoints-Skeleton-Shape** prediction manner. Specifically, our method completes missing topology using three steps: **1) Aligned Keypoint Localization.** An asymmetric keypoint locator, including an unsupervised multi-scale keypoint detector and a complete keypoint generator, is proposed for localizing aligned keypoints from complete and partial point clouds. We theoretically prove that the detector can capture aligned keypoints for objects within a sub-category. **2) Surface-skeleton Generation.** A new type of skeleton, named Surface-skeleton, is generated from keypoints based on geometric priors to fully represent the topological information captured from keypoints and better recover the local details. **3) Shape Refinement.** We design a refinement subnet where multi-scale surface-skeletons are fed into each recursive skeleton-assisted refinement module to assist the completion process. Experimental results show that our method achieves the state-of-the-art performance on point cloud completion.

1. Introduction

The geometry and vision community has put huge effort into point cloud processing, which is challenging due to the unordered, unstructured characteristics, and complex semantics of the point clouds. However, in real applications, occlusions and insufficient lighting lead to partial scans of real shapes and degrade the performance of subsequent processing. Point cloud completion focuses on predicting miss-

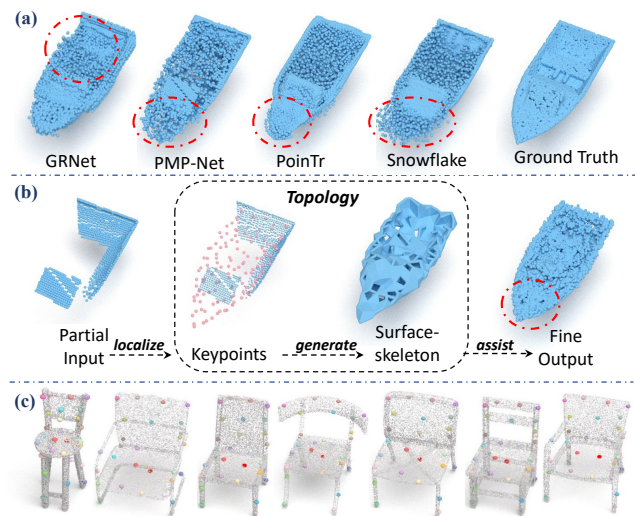


Figure 1. Illustration of (a) visual comparison results of current completion methods, (b) completion process of our LAKe-Net, and (c) aligned keypoints. Compared with GRNet [34], PMP-Net [29], PoinTr [40] and Snowflake [32], LAKe-Net can effectively recover missing topology part (see Red ellipses).

ing regions from partial observations, and shows its unique significance in many fundamental applications.

Recent works [7, 18, 26, 29, 32, 34, 35, 40, 41] for point cloud completion successfully utilized deep-learning methods and achieved more plausible and flexible results compared with traditional geometric-based methods [8, 19, 23] and alignment-based methods [11, 12, 16]. However, most existing methods directly predict the location of complete points, without predicting structured and topological information of the complete shape, which leads to coarse results in missing regions (see Figure 1(a)).

Inspired by typical geometric modeling theory that a complete 3D object includes geometry and topology, e.g., coordinates and connectivity, we tend to predict both geometric and structured topological information for point cloud completion, including keypoints and generated skeleton. To this end, we propose a novel **Keypoints-Skeleton-Shape** prediction manner, including three steps: keypoint

*Corresponding authors.

localization, skeleton generation, and shape refinement.

We firstly introduce the keypoint localization. Different from down-sampled points, keypoints are evenly distributed across semantic parts of the shape, and are considered as a crucial representation of geometric structure which are widely-used in many vision applications [17,27,37]. Therefore, we hold the belief that *once the keypoints and their connectivity are correctly localized, the entire geometry is determined*. To this end, we wish to localize complete keypoints according to partial inputs under the supervision of the ground truth keypoints. However, obtaining keypoints annotation for a large number of 3D data is difficult and expensive, so we propose an asymmetric keypoint locator including an unsupervised multi-scale keypoint detector (UMKD) and a complete keypoint generator (CKG) for complete and partial point cloud, respectively. For UMKD, we extract *Aligned Keypoints*, which means the order of keypoints are the same on different objects within a certain category (Figure 1(c)), so as to represent more stable and richer information, and provide stronger supervision for predicting complete keypoints from partial inputs by CKG.

Since discrete and sparse keypoints are not enough for representing the whole objects, we leverage skeleton to better represent topological details. Inspired by existing skeleton extraction methods [1, 5, 14], we propose a novel *Surface-skeleton*, which is generated from keypoints based on geometric priors. Compared with other types of skeletons, our surface-skeleton is a mixture of curves and triangle surfaces, and can represent more complex shape information. We integrate surface-skeletons with different fineness generated by multi-scale keypoints into the shape refinement step to recover finer results. Specifically, we propose a folding-based refinement subnet including three recursive skeleton-assisted refinement modules (RSR) following some of other completion methods [28,32].

In general, we propose LAKe-Net, a novel topology-aware point cloud completion model by localizing aligned keypoints. The whole pipeline includes four parts: auto-encoder, asymmetric keypoints locator, surface-skeleton generator and the refinement subnet. We leverage pairs of complete and partial point cloud during training. In detail, the input point clouds (either complete or partial) are firstly fed into an auto-encoder to learn a feature embedding space and generate coarse and complete results. Then, we localize multi-scale keypoints using asymmetric keypoints locator and generate corresponding surface-skeletons. The multi-scale structures are fed into the refinement subnet to generate fine output. The training process includes two stages for point cloud reconstruction and completion, respectively.

Overall, we summarize our main contributions as follows: (1) We propose LAKe-Net, a novel topology-aware point cloud completion model that utilizes a structured representation of the surface as assistance, including Aligned

Keypoints and Surface-skeleton, with a new Keypoints-Skeleton-Shape prediction manner. (2) We introduce an asymmetric keypoint locator including an unsupervised multi-scale keypoint detector and a complete keypoint generator, which can capture accurate keypoints for complete and partial objects in multiple categories, respectively. We theoretically prove that our detector detects aligned keypoints within each sub-category. (3) We conduct point cloud completion experiments on two datasets, PCN and ShapeNet55. Experimental results show that our LAKe-Net achieves the state-of-the-art performance on both datasets.

2. Related Works

Point Cloud Completion. Point cloud completion focuses on predicting missing shapes from partial point cloud input. Recently, inspired by point cloud analysis approaches [20, 21], PCN [41] first adopts an encoder-decoder architecture and a coarse-to-fine manner to generate the complete shape. Several works [15, 26, 32, 43] follow this practice and make modifications in network structure to obtain better performance. SA-Net [28] further extends the decoding process into multiple stages by introducing hierarchical folding. More recently, PoinTr [40] reformulates point cloud completion as a set-to-set translation problem and designs a new transformer-based encoder-decoder for point cloud completion. However, these methods mostly predict the location of complete points without predicting structured and topological information, which leads to coarse results in missing regions. SK-PCN [18] is the most relevant work to ours which pre-processes the dataset and uses meso-skeletons as supervision. However, SK-PCN doesn't predict the structured and topological information of original shape. Our proposed LAKe-Net utilizes aligned keypoints and corresponding surface-skeleton which can capture the shared topological information as an assistant for completion, and obtains better performance.

Skeleton Representation. Skeleton representation is widely-used in motion recognition [38,39], human pose estimation [2, 22] and human reconstruction [4, 42]. Jiang *et al.* [10] propose to incorporate skeleton awareness into the deep learning-based regression for 3D human shape reconstruction from point clouds. Tang *et al.* [25] utilize topology preservation property of skeleton to perform 3D surface reconstruction from a single RGB image. P2P-Net [36] learns bidirectional geometric transformations between point-based shape representations from two domains, surface-skeletons and surfaces. Our method designs surface-skeleton representations generated by multi-scale keypoints in a more fine-grained manner to progressively aid point cloud completion.

Unsupervised keypoint detection. While most hand-crafted 3D keypoint detectors fail to detect accurate and well-aligned keypoints in complex objects, Li *et al.* [13]

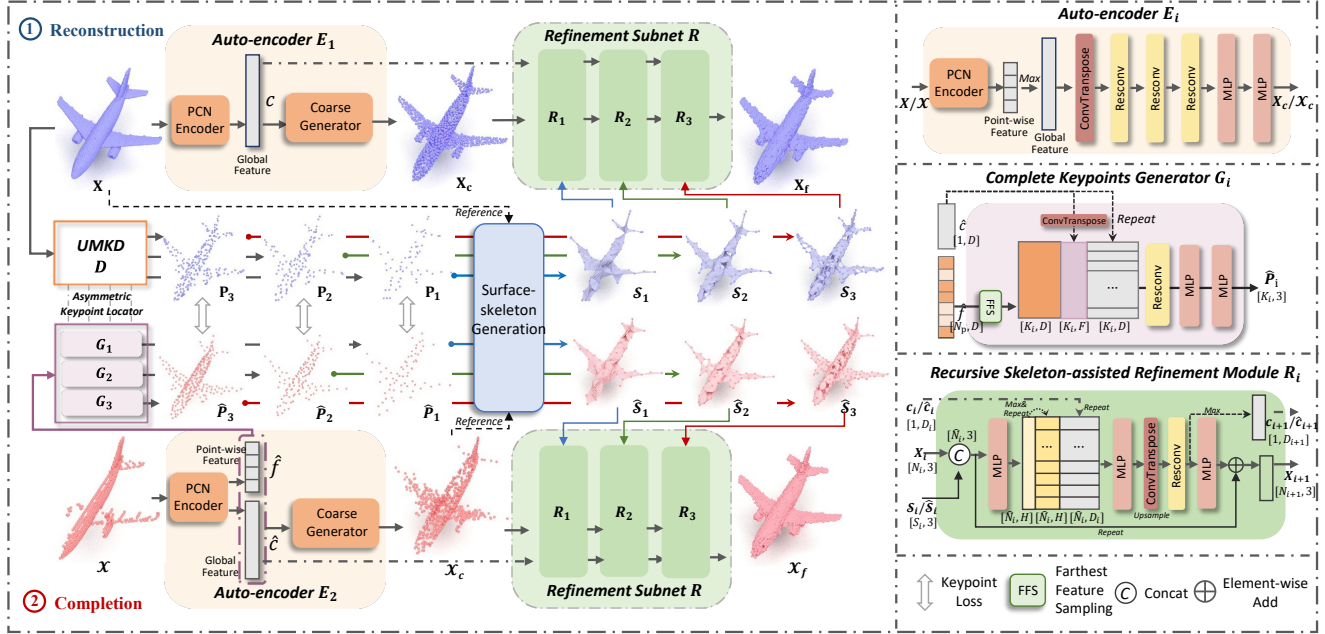


Figure 2. The overall architecture of LAKe-Net, which consists of two parts including Point Cloud Reconstruction (Blue) and Point cloud Completion (Red). We show the detailed structure of (a) Auto-encoder E , (b) Complete Keypoint Generator G and (c) Recursive Skeleton-assisted Refinement module R on the right side. PCN encoder is firstly proposed in [41]. UMKD denotes the unsupervised multi-scale keypoint detector. Surface-skeletons \mathcal{S}_i and $\hat{\mathcal{S}}_i$ are generated by \mathbf{P}_i and $\hat{\mathbf{P}}_i$, respectively.

propose the first learning-based 3D keypoint detector USIP. However, the detected keypoints are neither ordered nor semantically salient. Fernandez *et al.* [6] utilize symmetry prior in point clouds to capture keypoints in an unsupervised manner. Recently, Jakab *et al.* [9] further explore the application of unsupervised keypoints in shape deformation task. SkeletonMerger [24] proposes a novel keypoint detector based on an autoencoder architecture. However, different models need to be trained for different categories. We propose UMKD, an unsupervised multi-scale keypoint detector which can capture keypoints for objects in multiple categories. We find that it reaches the best performance on categories with shared topology on KeypointNet [37] and produces more salient and semantic richer keypoints.

3. Proposed Method

We propose a novel topology-aware point cloud completion network by localizing aligned keypoints (LAKe-Net), whose overall architecture is shown in Fig. 2. The pipeline includes four parts: auto-encoder, asymmetric keypoint locator, surface-skeleton generation and shape refinement. The training includes two stages: point cloud reconstruction and completion. The training data consists of pairs of complete and partial point clouds $(\mathbf{X}, \mathcal{X})$, where $\mathbf{X} \in \mathbb{R}^{N_c \times 3}$ and $\mathcal{X} \in \mathbb{R}^{N_p \times 3}$ denote the coordinates of complete and partial point clouds, N_c and N_p denote the number of points of complete and partial data, respectively.

Firstly, see upper part of Fig. 2, we utilize complete data \mathbf{X} as input to train an unsupervised multi-scale keypoint detector (UMKD) D , which extracts the multi-scale keypoints

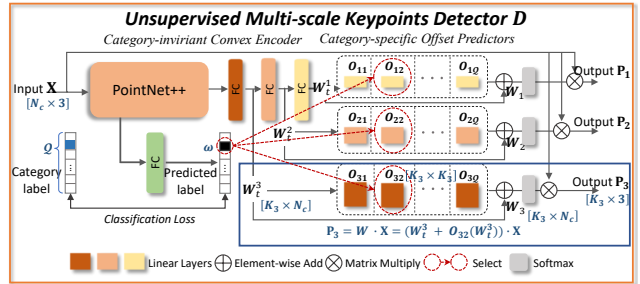


Figure 3. The detailed structure of our proposed UMKD. Take \mathbf{P}_3 as an example, we show the calculation process and dimensions of relative tensors in Blue.

$\{\mathbf{P}_i\}_{i=1}^3$, and an auto-encoder E_1 , which maps the inputs into a global feature space c and obtain coarse results \mathbf{X}_c . Then, a surface-skeleton generation process is employed to leverage the topology information in keypoints to construct a finer representation. Finally, a refinement subnet adopts the topology information \mathcal{S} and coarse results \mathbf{X}_c to generate high-resolution results \mathbf{X}_f . The reconstruction part unsupervisedly trains the UMKD to learn aligned keypoints for later completion stage and also provides a good initialization of the network to predict complete shape.

In the second stage, we fix the weights of keypoint detector D and auto-encoder E_1 , then input partial data \mathcal{X} into a new auto-encoder E_2 and predict complete coarse results \mathcal{X}_c . It is noteworthy that the auto-encoder E_1 and E_2 have the same architecture which includes a PCN encoder [41] and coarse point generator. The encoder of E_2 embeds in-

put points into point-wise local features $\hat{f} \in \mathbb{R}^{N_p \times d}$, where d denotes the dimension of feature embedding. We consider the maximum value $\hat{c} = \max_{N_p}(\hat{f}) \in \mathbb{R}^{1 \times d}$ as global features. Then we fuse the local and global features and feed the fused feature into a complete keypoint generator (CKG) G to predict multi-scale keypoints $\{\hat{\mathbf{P}}_i\}_{i=1}^3$, supervised by keypoints $\{\mathbf{P}_i\}_{i=1}^3$ detected in the first stage. At last, we send \mathcal{X}_c and interpolated surface-skeleton $\hat{\mathcal{S}}$ to a skeleton-assisted refinement subnet R to generate fine outputs \mathcal{X}_f . We describe the technical details of our proposed modules and training losses in the following sections.

3.1. Unsupervised Multi-scale Keypoint Detector

Given a set of N_c input complete point clouds $\mathbf{X} = \{x_j | j = 1, \dots, N_c\} \in \mathbb{R}^{N_c \times 3}$, our aim is to predict $\{K_i\}_{i=1}^3$ numbers of keypoints $\{\mathbf{P}_i\}_{i=1}^3 = \{p_k | k = 1, \dots, K_i\} \in \mathbb{R}^{K_i \times 3}$. Specifically, we tend to predict convex combination weights $\mathbf{W}_i = \{w_{ij}\} \in \mathbb{R}^{N_c \times K_i}$ of point clouds instead of predicting the coordinates of keypoints directly to avoid deviating from the original shape. So the predicted keypoints \mathbf{P}_i are derived by:

$$\mathbf{P}_i = \mathbf{W}_i^T \mathbf{X} = \sum_{j=1}^{N_c} w_{ij}^T x_j, s.t., w_{ij} > 0, \sum_{j=1}^{N_c} w_{ij} = 1. \quad (1)$$

To predict convex weight \mathbf{W}_i using a single model for all categories which adapts to our pipeline, we assume $\mathbf{W}_i = \mathbf{W}_t^i + \mathbf{W}_o^i$, where \mathbf{W}_t^i denotes a category-invariant template weight and \mathbf{W}_o^i denotes a category-specific weight offset. Besides, we expect to predict multi-scale keypoints for subsequent tasks. To this end, we propose an Unsupervised Multi-scale Keypoint Detector (UMKD) which consists of a category-invariant convex encoder and category-specific offset predictors. The detailed structure of UMKD is shown in Figure 3. In detail, we apply a PointNet++ [21] as a backbone encoder to extract local and global features. It includes four set abstraction layers to group and down-sample input points. Then the global and local features are propagated back to each partial point. Then we input point-wise features into three fully connected blocks and extract multi-scale convex features $\mathbf{W}_t^i \in \mathbb{R}^{K_i \times N_c}$ progressively.

As for predicting category-specific offset \mathbf{W}_o^i , we firstly predict the category label ω for input shape and send the \mathbf{W}_t^i to the relative offset predictor $\mathbf{O}_{i\omega}$. We add two fully-connected layers after the last pointset abstraction layer in PointNet++ as classification head to predict the category label of every input geometries. $\mathbf{O}_{ij} \in \mathbb{R}^{K_i \times K_i}$ is a learnable matrix. We set $j \in [1, Q]$. Q denotes the number of categories. Take \mathbf{P}_3 as an example (shown in Figure 3), the input points $\mathbf{X} \in \mathbb{R}^{N_c \times 3}$ are send to the convex encoder and output template convex weights $\mathbf{W}_t^3 \in \mathbb{R}^{K_3 \times N_c}$ and predicted label $\omega = 2$. Then \mathbf{W}_t^3 are input to the selected offset predictor \mathbf{O}_{32} and output $\mathbf{W}_o^3 \in \mathbb{R}^{K_3 \times N_c}$. At last, we get

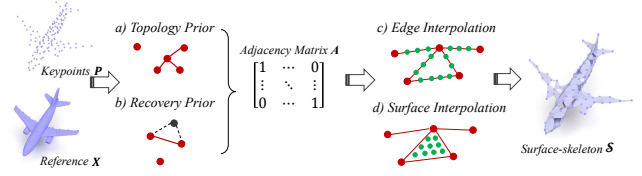


Figure 4. The surface-skeleton generation operation.

the final convex weight $\mathbf{W}_3 = \mathbf{W}_t^3 + \mathbf{W}_o^3$ and normalize \mathbf{W}_3 by a softmax function and predict keypoint \mathbf{P}_3 .

The keypoints extracted by UMKD D follow a theory that: *The coordinates of detected keypoints P are irrelevant to the order of original points X . That is: $P = D(X) = D(\mathcal{R}(X))$, where $\mathcal{R}(\cdot)$ denotes random permutation operation.* The theoretical proof is introduced in Supplementary Materials. Therefore, the detected keypoints are aligned among objects with shared topology within a sub-category (shown in Figure 1(c)).

3.2. Surface-skeleton Generation

After extracting the keypoints of original point clouds, our aim is to reconstruct the point cloud according to the extracted keypoints. We consider skeleton as an intermediate representation between keypoints and original point clouds. Previous methods like SkeletonMerger [24] used skeletons connected by every pair of keypoints, which leads to high computational complexity and a lot of invalid points. Moreover, given a surface, previous skeletons either are located near the medial axis of the surface [5], or are mostly located outside the surface [24]. We aim at extracting a skeleton that are located near the surface, so that it can better assist the later refinement process. In order to represent both topological and geometric information of a complex shape, we design a surface-skeleton structure which is generated by keypoints and consists of a mixture of curves and triangle surfaces adapted to the local 3D geometry. Specifically, given predicted keypoints $P \in \mathbb{R}^{K \times 3}$ and reference points X , we follow the two shape priors proposed in [14] and generate the skeletal graph: (1) the topology prior that each node has links to its top-2 nearest nodes; (2) the recovery prior that two keypoints are linked if they are two nearest keypoints of a reference point.

After skeletal graph generation, we can get an adjacency matrix $A \in \mathbb{R}^{K \times K}$. We propose a surface interpolation strategy based on Delaunay-based triangulation region. The whole interpolation process consists of two steps: edge interpolation and triangle surface interpolation. Given a certain graph, we firstly interpolate points into each connected edge. Next, we detect every triangle surface according to the skeletal graph. Then we insert points into the triangle region. The number of interpolated points is proportional to the triangle area. The overall skeletal graph generation and surface interpolation process is shown in Fig 4. It is noteworthy that we utilize the complete point clouds \mathbf{X} as

reference shape for reconstruction while using the coarse results \mathcal{X}_c for completion.

Overall, we get the surface-skeleton \mathcal{S} . Increasing the number of keypoints leads to more complex surface-skeletons which can represent finer shape details. We utilize the structure as topology representation of geometric shape which is crucial for reconstruction and completion.

3.3. Complete Keypoint Generator

In the second stage, given the partial input $\mathcal{X} \in \mathbb{R}^{N_p \times 3}$, the aim of our proposed Complete Keypoint Generator (CKG) module is to predict multi-scale complete keypoints from partial feature embedding. To this end, we utilize the local and global feature \hat{f} and \hat{c} extracted by the encoder in E_2 as input. Similar to Farthest Point Sampling (FPS) strategy for point cloud, in order to downsample the point features, we use Farthest Feature Sampling (FFS) strategy to down sample point-wise local feature \hat{f} to sparse feature \hat{f}^* where we replace point coordinates in FPS by feature embeddings. The number of sampling is the same as predicted keypoints. Then we utilize a de-convolution layer to upsample the global feature \hat{c} and fuse them into a residual block and predict final keypoints. We train three similar blocks for multi-scale keypoints prediction. Then we generate corresponding surface-skeleton $\hat{\mathcal{S}}$ by surface interpolation introduced on Sec. 3.2.

3.4. Recursive Skeleton-assisted Refinement

Our proposed shape refinement subnet R includes three recursive skeleton-assisted refinement (RSR) modules that aim to integrate multi-scale surface-skeletons and coarse output from previous auto-encoder to predict finer geometric details in a recursive way. The detailed design of the module is shown in Fig 2. It follows existing methods [28, 32, 35] using a coarse-to-fine strategy to learn the offset of integrated points. Specifically, we progressively concatenate coarse point clouds obtained from previous steps and surface-skeleton generated by corresponding keypoints described in Sec. 3.2. We denote the input coarse points as $\mathbf{X}_{i-1} = \{x_j\}_{j=1}^{N_{i-1}}$ and surface-skeleton as $\mathcal{S}_{i-1} = \{p_j\}_{j=1}^{S_{i-1}}$. The integrated points $\hat{\mathbf{X}}_{i-1} = \text{concat}(\mathbf{X}_{i-1}, \mathcal{S}_{i-1})$ on the i -th step where $\text{concat}(\cdot)$ refers to concatenate operation. N_{i-1} and S_{i-1} denote numbers of coarse points and surface-skeleton, respectively. In this paper, we set $N_i = S_i$ in each step. Therefore, the updated points output by the i -th RSR module $\mathbf{X}_i = \hat{\mathbf{X}}_{i-1} + R(\hat{\mathbf{X}}_{i-1})$ will be sent to the next step.

3.5. Training and Losses

Point Cloud Reconstruction. In the first stage, the UMKD D , auto-encoder E_1 and refinement subnet R are trained together. The training losses are divided into two parts, one is to constrain the keypoint detection, the other is data

reconstruction. Firstly, in order to encourage the detected keypoints \mathbf{P} to be well-distributed and not deviate from the global shape, we calculate the Chamfer Distance (CD) loss between the predicted keypoints and sparse point clouds \mathbf{X}^* downsampled from input data using FPS strategy. As for training one detector within several categories, we also train a classification head. We denote the predicted output is ω and certain category label is σ . We train a criterion loss \mathcal{L}_{cls} . Besides, as mentioned in Sec. 3.2, we expect the surface-skeleton can reconstruct the geometric shape of the ground truth. We calculate CD between multi-scale surface-skeletons $\{\mathcal{S}_i\}_{i=1}^3$ and the ground truth \mathbf{X} . So the overall loss for training keypoint detector is:

$$\mathcal{L}_{CD} = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \|x - y\|^2 + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} \|y - x\|^2, \quad (2)$$

$$\mathcal{L}_{cls} = - \sum_{i=1}^Q (\sigma_i \log \omega_i + (1 - \sigma_i) \log(1 - \omega_i)), \quad (3)$$

$$\mathcal{L}_{kp} = \mathcal{L}_{CD}(\mathbf{P}, \mathbf{X}^*) + \sum_{i=1}^3 \mathcal{L}_{CD}(\mathcal{S}_i, \mathbf{X}) + \mathcal{L}_{cls}. \quad (4)$$

At last, we calculate CD between the ground truth \mathbf{X} and sparse output \mathbf{X}_c , dense output \mathbf{X}_f , respectively.

$$\mathcal{L}_{rec} = \mathcal{L}_{CD}(\mathbf{X}_c, \mathbf{X}) + \mathcal{L}_{CD}(\mathbf{X}_f, \mathbf{X}). \quad (5)$$

In general, the overall training loss in the first stage is:

$$\mathcal{L}_1 = \mathcal{L}_{rec} + \lambda_{kp}^1 \mathcal{L}_{kp}, \quad (6)$$

where λ_{kp}^1 denotes hyper-parameters to balance inference.

Point Cloud Completion. In the second stage, we fix the weights of UMKD D and auto-encoder E_1 , and train a new auto-encoder E_2 and CKG G . The refinement subnet R pre-trained before continues to be optimized. We constrain keypoints prediction using absolute distance between predicted keypoints $\hat{\mathbf{P}}$ and ground truth keypoints \mathbf{P} :

$$\mathcal{L}_{kp}^c = \sum_{i=1}^3 \sum_{j=1}^{K_i} \|p_{ij} - \hat{p}_{ij}\|^2. \quad (7)$$

Same as other concurrent network [31], we align global features c and $\hat{c} \in \mathbb{R}^{1 \times d}$ encoded by auto-encoders in two stages for hidden feature space learning:

$$\mathcal{L}_{feat} = \frac{1}{d} \sum_{i=1}^d \|c_i - \hat{c}_i\|^2. \quad (8)$$

As for typical training on completion task, we follow the coarse-to-fine process in the first stage. The coarse output \mathcal{X}_c and fine output \mathcal{X}_f are optimized using CD loss:

$$\mathcal{L}_{com} = \mathcal{L}_{CD}(\mathcal{X}_c, \mathbf{X}) + \mathcal{L}_{CD}(\mathcal{X}_f, \mathbf{X}). \quad (9)$$

In the summary, the full objective of point cloud completion in the second stage is:

$$\mathcal{L}_2 = \mathcal{L}_{com} + \lambda_{kp}^2 \mathcal{L}_{kp}^c + \lambda_{feat} \mathcal{L}_{feat}, \quad (10)$$

where $(\lambda_{kp}^2, \lambda_{feat})$ denote hyper-parameters.

4. Experiments

4.1. Dataset Setting and Evaluation metric

PCN: The PCN dataset is a widely-used benchmark for point cloud completion, which is created by [41], including different objects from 8 categories: plane, cabinet, car, chair, lamp, sofa, table, and vessel. The training set contains 28,974 objects, while validation and test set contains 800 and 1,200 objects, respectively. The complete point cloud consists of 16,384 points which are uniformly sampled on the original CAD model. Partial point cloud, consisting of 2,048 points, is created by back-projecting 2.5D depth images into 3D from 8 random viewpoints.

ShapeNet55: To explore the performance of our method on a large number of categories, we evaluate our method on all 55 categories of ShapeNet [3], named ShapeNet55. The ShapeNet55 dataset was first created by PoinTr [40]. The training set contains 41,952 objects, while test set contains 10,518 objects. We randomly sample 80% objects in each category to form training set and use the rest 20% to form validation set.

Evaluation Metrics: We utilize two evaluation metrics between output point cloud and the ground truth, Chamfer Distance (CD) using L2 norm and Earth Mover’s Distance (EMD), following most of the methods on PCN and ShapeNet55 test set. CD is introduced in Equation 2 and EMD is defined as:

$$EMD(X, Y) = \min_{\phi: X \rightarrow Y} \frac{1}{|X|} \sum_{x \in X} \|x - \phi(x)\|_2, \quad (11)$$

where ϕ is a bijection. It is noteworthy that we compute these metrics using 16,384 and 8,192 points for PCN and ShapeNet55, respectively.

4.2. Implementation Details

The whole training of LAKE-Net is a two-stage process: point cloud reconstruction and point cloud completion. The input of the first stage (reconstruction) is a set of complete point clouds with coordinates and object category labels from training set of all datasets. We train the keypoint detection for 60 epochs and progressively extract 256, 128, 64 keypoints. The refinement subnet includes three RSR modules, the up factors of de-convolution are [1,1,2]. For the second stage (the bottom completion branch of Fig. 2), we only input partial point clouds from training set with its coordinate information. We utilize Adam optimization to train the whole architecture of point cloud completion for 100 epochs with batchsize 64 and learning rate 0.001. The hyper-parameters $\lambda_{kp}^1 = \lambda_{kp}^2 = 10$, $\lambda_{feat} = 1000$. The inference time of our method is 34.5ms per sample.

4.3. Results on PCN dataset.

We compare the performance of our proposed LAKE-Net and other state-of-the-art completion methods. We im-

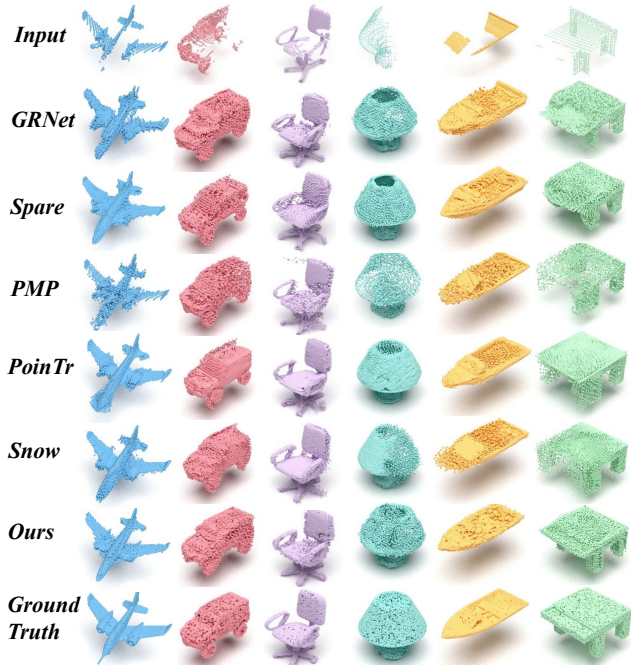


Figure 5. Visualization of point cloud completion comparison results on PCN dataset with other recent methods.

plement other methods using their open source code and hyper-parameters for fair comparison. Table 2 and 3 show the quantitative comparison results of our method and other point cloud completion methods on PCN datasets, from which we can see that our method achieves the best performance over all counterparts on both CD and EMD metrics. Specifically, compared with the second-ranked Snowflake which also proposed progressive decoding modules, our method has better performance with the help of aligned keypoints and surface-skeletons. Besides, according to experimental results, our proposed LAKE-Net is more powerful to predict symmetrical geometries and their topology information compared with SnowflakeNet.

Moreover, we also show the visualization of qualitative comparison results and some recent methods in Figure 5, which show that our method has better performance on completing missing topology. Specifically, methods which also utilize progressive coarse-to-fine decoding like PMP-Net and SnowflakeNet, tended to predict coarse missing shape and generate scattered points, especially for geometry with a plane or surface. Other methods like GRNet, SpareNet and PoinTr are weak on recovering the local details and some missing topology like table legs. Our method can predict geometries with more clear topology structure and fewer noises.

4.4. Results on ShapeNet55 dataset

Moreover, to evaluate the generalization and powerful of our method on a large account of categories of data

Category	Bed		Bench		Bookshelf		FileCabinet		Faucet		Telephone		Can		Flowerpot		Tower		Pillow		Average	
Metrics	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD	CD	EMD
Folding [35]	3.17	73.6	1.45	50.1	2.48	64.4	1.94	65.3	3.19	66.2	0.69	39.1	1.76	60.2	4.11	82.9	1.83	59.5	1.64	63.2	2.06	60.2
PCN [41]	2.50	49.4	0.96	28.9	2.39	44.7	1.49	37.2	1.96	40.3	0.54	24.0	1.30	30.9	2.58	48.7	1.34	33.7	1.09	31.5	1.36	34.0
GRNet [34]	0.93	29.7	0.86	25.8	0.93	29.6	1.57	24.9	0.83	27.6	0.87	26.2	1.15	32.3	1.24	33.5	0.87	25.3	1.06	28.8	1.15	28.2
PoinTr [40]	2.18	37.6	0.93	21.4	1.86	37.1	3.23	42.7	1.75	42.4	0.55	20.8	2.13	31.2	2.68	42.7	1.73	35.9	1.40	31.8	1.70	31.7
Ours	0.72	28.4	0.71	18.2	0.89	29.7	0.97	16.4	0.34	20.5	0.48	20.8	0.63	29.0	1.19	35.9	0.60	21.9	0.97	29.8	0.89	31.0

Table 1. Quantitative comparison results with other completion methods on ShapeNet55 dataset using $CD-l_2(\times 10^3)$ and $EMD(\times 10^3)$ metrics. We report the detailed results for each method on 10 sampled categories and overall average results on all 55 categories.

$CD-l_2(\times 10^4)$	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel	Average
Folding [35]	3.151	7.943	4.676	9.225	9.234	8.895	6.691	7.325	7.142
PCN [41]	1.400	4.450	2.445	4.838	6.238	5.129	3.569	4.062	4.016
AtlasNet [7]	1.753	5.101	3.237	5.226	6.342	5.990	4.359	4.177	4.523
MSN [15]	1.543	7.249	4.711	4.539	6.479	5.894	3.797	3.853	4.758
GRNet [34]	1.531	3.620	2.752	2.945	2.649	3.613	2.552	2.122	2.723
PMP-Net [29]	1.205	4.189	2.878	3.495	2.178	4.267	2.921	1.894	2.878
SpareNet [33]	1.756	6.635	3.614	6.163	6.313	7.893	4.987	3.835	5.149
PointTr [40]	0.993	4.809	2.529	3.683	3.077	6.535	3.103	2.029	3.345
Snowflake [32]	0.913	3.322	2.246	2.642	1.898	3.966	2.011	1.692	2.336
Ours	0.646	2.594	1.743	2.149	2.759	2.186	1.876	1.602	1.944

Table 2. Quantitative comparison results with other methods of point cloud completion on PCN using $CD-l_2$ (lower is better).

$EMD(\times 10^2)$	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel	Average
Folding [35]	1.682	2.576	2.183	2.847	3.062	3.003	2.500	2.357	2.526
PCN [41]	2.426	1.888	2.744	2.200	2.383	2.062	1.242	2.208	2.144
AtlasNet [7]	1.324	2.582	2.085	2.442	2.718	2.829	2.160	2.114	2.282
MSN [15]	1.334	2.251	2.062	2.346	2.449	2.712	1.977	2.001	2.142
GRNet [34]	1.376	2.128	1.918	2.127	2.150	2.468	1.852	1.876	1.987
PMP-Net [29]	1.259	2.058	2.520	1.798	1.280	2.579	1.651	1.760	1.863
SpareNet [33]	1.131	2.014	1.783	2.050	2.063	2.333	1.729	1.790	1.862
PointTr [40]	0.938	1.986	1.851	1.892	1.740	2.242	1.931	1.532	1.764
Snowflake [32]	1.375	2.633	2.591	2.086	1.599	3.070	1.616	1.957	2.116
Ours	0.958	1.830	1.564	1.667	1.782	1.755	1.499	1.402	1.557

Table 3. Quantitative comparison results with other methods of point cloud completion on PCN using EMD (lower is better).

to adapt to real-world scenarios, we conduct experiments on ShapeNet55 dataset and compare with other completion methods. We drop 75% of the complete point cloud and re-sample the remaining partial point clouds to 2,048 points as input for all methods. Table 1 shows the quantitative comparison results on 10 sampled categories. The last column shows the overall average results of 55 categories. We can see that our method achieves the best result on CD metric and have competitive results on EMD metric. Specifically, the results on *Bed*, *Bench*, *Bookshelf*, *FileCabinet*, *Faucet*, which are similar as samples in PCN datasets or have shared topology within a sub-category, show that our method can recover geometries more efficiently using topology assistant. Moreover, the results on other categories show that our method is more powerful in completing geometries with regular and symmetrical contours, similar as *Vessel* in PCN dataset. We visualize the completion process of our method on samples from ShapeNet55 in Figure 6. It can be seen that our method can localize effect keypoints and recover missing topological and geometric information with the help of surface-skeletons.

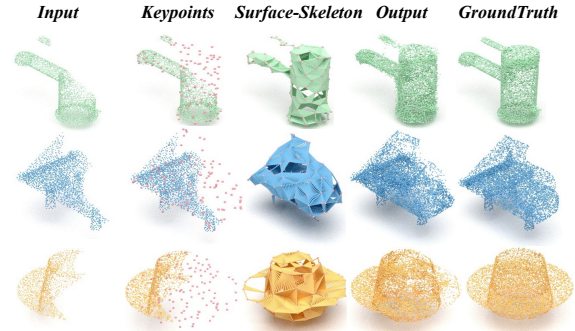


Figure 6. Visualization of completion on ShapeNet55 dataset by our proposed method. We also show the predicted keypoints and generated surface-skeletons on the second and third columns.

5. Method Analysis

In this section, we examine the effectiveness of our motivations in LAKE-Net. We conduct several ablation studies from different points of view. For fair comparison, all methods are trained and tested on PCN dataset for completion and KeypointNet for keypoint detection.

Unsupervised Multi-scale Keypoint Detector. To prove the effectiveness and accuracy of our proposed UMKD, we evaluate our extracted keypoints compared with two recent unsupervised keypoint detectors Fernandez *et al.* [6] and SkeletonMerger [24]. All methods are trained and tested on KeypointNet [37], which has keypoints annotations with semantic correspondence labels. We evaluate these methods on five categories: airplane, car, chair, table and vessel. Specifically, we detect 16, 32, 64 keypoints for all categories. As for other methods, we train five models and detect 16 keypoints for each category. We firstly down-sample the same number of predict keypoints as the annotated keypoints using the nearest neighbor strategy.

For evaluation metrics, we follow SkeletonMerger and utilize mean Intersection over Unions (mIoU) metrics to evaluate the keypoint silence and accuracy. It is calculated with a threshold of 0.1 using euclidean distance. The quantitative results are shown in Table 4, which illustrates that our proposed keypoint detector trained on multiple categories has competitive performance, or even better, than other unsupervised methods trained on a single category. The results on *airplane*, *chair* and *table* also show that our method has better performance on geometries with obvious topological structures. The visualization is shown in Fig-

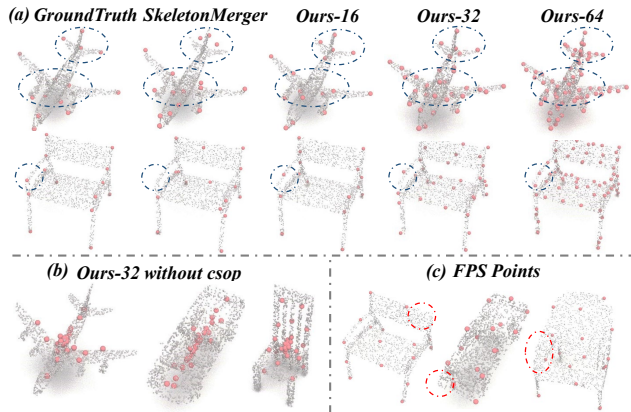


Figure 7. Visualization of (a) multi-scale keypoint detected by ours and SkeletonMerger on KeypointNet dataset; (b) failed case by our method without *csop*; (c) drawbacks of FPS 16 points. *csop* denotes category-specific offset predictors.

mIoU	Airplane	Car	Chair	Table	Vessel
Fernandez <i>et al.</i>	69.7	50.5	51.2	49.3	53.5
SkeletonMerger	72.7	64.6	63.2	59.6	62.0
Ours-16	73.2	58.1	69.2	62.5	61.3
Ours-32	73.7	60.2	70.5	63.2	62.5
Ours-64	74.0	62.9	71.3	65.4	64.0
Ours-32 <i>w/o csop</i>	35.4	13.2	27.8	23.9	11.0

Table 4. Quantitative comparison results with other unsupervised keypoint detector on KeypointNet using mIoU (higher is better).

ure 7(a). Our detector can produce more salient and semantic richer keypoints. We also show our detected keypoints in multi-scale which represent finer geometric details.

Besides, to evaluate the effectiveness of our proposed category-specific offset predictors (*csop*) in UMKD, we replace all offset predictors with a single category invariant offset predictor. The qualitative results is visualized in Figure 7(b) and quantitative results are shown in Table 4. It is obvious that a single offset predictor cannot handle geometries in multiple categories. The predicted keypoints tend to aggregate together to reduce the loss.

Keypoints and Surface-skeletons. To evaluate the necessity of using aligned keypoints and surface-skeletons for point cloud completion, we conduct several ablated experiments. We consider the auto-encoder E_2 and refinement subnet R in the second stage as baseline. In particular, we replace multi-scale complete keypoints detected in the first stage with multi-scale down-sampled points from ground truth using FPS strategy. And we use CD loss between down-sampled points and predicted keypoints in the second stage. We also remove the assist of generated surface-skeletons and change the up factors into [2,2,4] for fair comparison. The quantitative results are illustrated in Table 5. We can see that the down-sampled points cannot represent efficient topology information (as shown in Figure 7(c)) especially on some joint parts, and are not helpful for completing missing geometries in our pipeline. Besides,

EMD($\times 10^2$)	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel	Average
Ours	0.958	1.830	1.564	1.667	1.782	1.755	1.499	1.402	1.557
-use FPS	1.117	2.295	1.978	2.157	1.916	2.607	1.810	1.823	1.963
-w/o S-sk	1.469	2.638	2.386	2.380	2.221	2.989	1.906	2.020	2.251
PointDisturb	1.031	1.902	1.554	2.012	1.945	2.037	1.684	1.437	1.700
ClassDisturb	0.963	1.846	1.576	1.786	1.831	1.780	1.545	1.397	1.590

Table 5. Ablation studies and robustness test on PCN dataset using EMD metrics. w/o S-sk denotes without surface-skeleton.

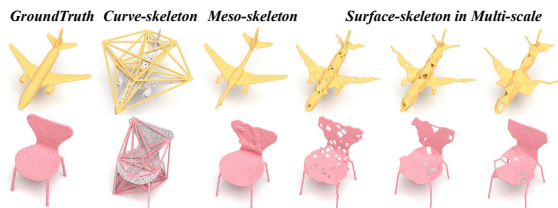


Figure 8. Visualization of different type of skeletons, including curve skeleton generated by [24], meso-skeleton from [30] and our surface-skeleton in multi-scale.

CD loss between two unordered and sparse point clouds is harder to be optimized to convergence.

We also visualize different types of skeletons compared with our multi-scale surface-skeleton in Figure 8. It can be seen that our surface-skeleton focuses on representing surfaces of original shape, and can get competitive performance with meso-skeletons detected by typical method [30], and better than curve skeleton from [24].

Robustness Test. We also conduct ablation studies to investigate the robustness of our method in some extreme cases. We firstly randomly disturb 5% of the detected GT keypoints with a threshold of 0.1 after the first stage. Secondly, we deliberately misclassify geometries with similar shapes: table, chair and sofa. The results are reported on Table 5. Experimental results show that our method is robust to errors in keypoints detection in the first stage.

6. Conclusion

In this paper, we propose a novel topology-aware point cloud completion method, named LAKe-Net, which focuses on completing missing topology by localizing aligned keypoints, with a novel Keypoints-Skeleton-Shape prediction manner, including aligned keypoints localization, surface-skeleton generation and shape refinement. Experimental results show that our LAKe-Net achieves the state-of-the-art performance on point cloud completion.

7. Acknowledgements

This work was supported by the National Key Research and Development Program of China (2019YFC1521104), National Natural Science Foundation of China (72192821, 61972157, 62176092), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200, 22YF1420300, 21511100700), CAAI-Huawei MindSpore Open Fund, and Art major project of National Social Science Fund (18ZD22).

References

- [1] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su. Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference*, pages 187–197. IEEE, 2010. [2](#)
- [2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. [2](#)
- [3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [6](#)
- [4] K.-L. Cheng, R.-F. Tong, M. Tang, J.-Y. Qian, and M. Sarkis. Parametric human body reconstruction based on sparse key points. *IEEE transactions on visualization and computer graphics*, 22(11):2467–2479, 2015. [2](#)
- [5] N. D. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on visualization and computer graphics*, 13(3):530, 2007. [2](#), [4](#)
- [6] C. Fernandez-Labrador, A. Chhatkuli, D. P. Paudel, J. J. Guerrero, C. Démonceaux, and L. V. Gool. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 546–563. Springer, 2020. [3](#), [7](#)
- [7] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. [1](#), [7](#)
- [8] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):59–73, 2008. [1](#)
- [9] T. Jakab, R. Tucker, A. Makadia, J. Wu, N. Snavely, and A. Kanazawa. Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12783–12792, 2021. [3](#)
- [10] H. Jiang, J. Cai, and J. Zheng. Skeleton-aware 3d human shape reconstruction from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5431–5441, 2019. [2](#)
- [11] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31(4):1–11, 2012. [1](#)
- [12] V. G. Kim, W. Li, N. J. Mitra, S. Chaudhuri, S. DiVerdi, and T. Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013. [1](#)
- [13] J. Li and G. H. Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 361–370, 2019. [2](#)
- [14] C. Lin, C. Li, Y. Liu, N. Chen, Y.-K. Choi, and W. Wang. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4277–4286, 2021. [2](#), [4](#)
- [15] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. 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](#), [7](#)
- [16] A. Martinovic and L. Van Gool. Bayesian grammar learning for inverse procedural modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 201–208, 2013. [1](#)
- [17] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1584–1601, 2006. [2](#)
- [18] Y. Nie, Y. Lin, X. Han, S. Guo, J. Chang, S. Cui, and J. Zhang. Skeleton-bridged point completion: From global inference to local adjustment. In *Advances in Neural Information Processing Systems*, pages 16119–16130, 2020. [1](#), [2](#)
- [19] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number CONF, pages 23–32, 2005. [1](#)
- [20] C. R. Qi, H. Su, K. Mo, and L. 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. [2](#)
- [21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. [2](#), [4](#)
- [22] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018. [2](#)
- [23] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an rgb-d camera. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. [1](#)
- [24] R. Shi, Z. Xue, Y. You, and C. Lu. Skeleton merger: an unsupervised aligned keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 43–52, 2021. [3](#), [4](#), [7](#), [8](#)
- [25] J. Tang, X. Han, J. Pan, K. Jia, and X. Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4541–4550, 2019. [2](#)
- [26] L. P. Tchammi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. [1](#), [2](#)
- [27] H. Wang, J. Guo, D.-M. Yan, W. Quan, and X. Zhang. Learning 3d keypoint descriptors for non-rigid shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. [2](#)

- [28] X. Wen, T. Li, Z. Han, and Y.-S. Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1939–1948, 2020. 2, 5
- [29] X. Wen, P. Xiang, Z. Han, Y.-P. Cao, P. Wan, W. Zheng, and Y.-S. 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, 7
- [30] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6):1–13, 2015. 8
- [31] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla. Asfm-net: Asymmetrical siamese feature matching network for point completion. *arXiv preprint arXiv:2104.09587*, 2021. 5
- [32] P. Xiang, X. Wen, Y.-S. Liu, Y.-P. Cao, P. Wan, W. Zheng, and Z. 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, 5, 7
- [33] C. Xie, C. Wang, B. Zhang, H. Yang, D. Chen, and F. Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628, 2021. 7
- [34] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun. Grnet: Griding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020. 1, 7
- [35] Y. Yang, C. Feng, Y. Shen, and D. 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. 1, 5, 7
- [36] K. Yin, H. Huang, D. Cohen-Or, and H. Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 2
- [37] Y. You, Y. Lou, C. Li, Z. Cheng, L. Li, L. Ma, C. Lu, and W. Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656, 2020. 2, 3, 7
- [38] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 910–919, 2017. 2
- [39] T. Yu, Z. Zheng, K. Guo, J. Zhao, Q. Dai, H. Li, G. Pons-Moll, and Y. Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7287–7296, 2018. 2
- [40] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. 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, 6, 7
- [41] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1, 2, 3, 6, 7
- [42] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu. Deep network for the integrated 3d sensing of multiple people in natural images. *Advances in Neural Information Processing Systems*, 31:8410–8419, 2018. 2
- [43] W. Zhang, Q. Yan, and C. Xiao. Detail preserved point cloud completion via separated feature aggregation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 512–528. Springer, 2020. 2