Point Cloud Completion via Skeleton-Detail Transformer

Wenxiao Zhang, Huajian zhou, Zhen Dong, Jun Liu, Qingan Yan, Chunxia Xiao*

Abstract—Point cloud shape completion plays a central role in diverse 3D vision and robotics applications. Early methods used to generate global shapes without local detail refinement. Current methods tend to leverage local features to preserve the observed geometric details. However, they usually adopt the convolutional architecture over the incomplete point cloud to extract local features to restore the diverse information of both latent shape skeleton and geometric details, where long-distance correlation among the skeleton and details is ignored. In this work, we present a coarse-to-fine completion framework, which makes full use of both neighboring and long-distance region cues for point cloud completion. Our network leverages a Skeleton-Detail Transformer, which contains cross-attention and self-attention layers, to fully explore the correlation from local patterns to global shape and utilize it to enhance the overall skeleton. Also, we propose a selective attention mechanism to save memory usage in the attention process without significantly affecting performance. We conduct extensive experiments on the ShapeNet dataset and real-scanned datasets. Qualitative and quantitative evaluations demonstrate that our proposed network outperforms current state-of-the-art methods.

Index Terms—Point cloud, shape completion, point cloud completion

1 Introduction

As low-cost sensors like depth cameras and LIDAR are becoming increasingly available, 3D data has gained extensive attention in vision and robotics communities. However, viewpoint occlusion and low resolution in 3D scans always lead to incomplete shapes, which can not be directly used in practical applications. To this end, it is desired to recover complete 3D models from their partial ones, which have significant values in a variety of vision tasks [1], [2], [3], [4], [5], [6].

Early learning-based works succeed in performing shape completion on the volumetric representation of 3D objects, such as occupied grids or TSDF volume, where convolution operations can be applied directly [1], [7], [8], [9], [10]. However, volumetric representation always leads to expensive memory costs and low shape fidelity. In contrast, point cloud is a more compact representation of 3D data.

PCN [11] is the first learning-based work on point cloud completion. It recovers the completed 3D model via an embedded global feature vector, but fails to provide fine geometric details. Recently, some works [12], [13], [14], [15], [16] provide better completion results by preserving observed geometric details from the incomplete point shape using local features. However, they usually leverage convolutional operations to extract local features to restore the entire object shape, ignoring the long-distance correlation between the global skeleton and local patterns. As illustrated in Figure 1(a), since the four identical wheels are apart from each other, only aggregating local features can hardly leverage the similar semantic structure or symmetry prior from other wheels to constrain the shape completion and would lead to undesired distortions.

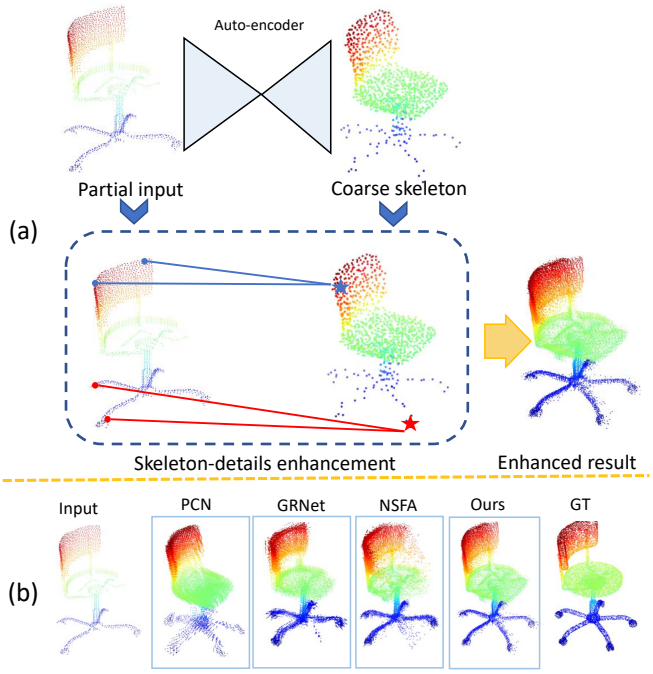


Fig. 1: (a) Given a partial input, the proposed network first reconstructs a coarse completion result and then enhances the coarse skeleton with neighboring and long-distance local patterns, as marked in blue and red lines. (b) Compared with current state-of-the-arts, our method performs better in both detail preservation and latent shape prediction.

- *W. Zhang, H. Zhou, and C. Xiao are with School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China. E-mail: wenxxiao.zhang@gmail.com, eagle_zhou@foxmail.com, cxxiao@whu.edu.cn*
- *D. Zhen is with State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China. E-mail: dongzhenwhu@whu.edu.cn*
- *J. Liu is with Singapore University of Technology and Design, 8 Somapah Rd, Singapore 487372. E-mail: jun_liu@sutd.edu.sg.*
- *Q.Yan is with InnoPeak Technology, Inc. 2479 E Bayshore Rd, Palo Alto, CA 94303. Email: yanqunganssg@gmail.com.*
- **Corresponding to Chunxia Xiao: cxxiao@whu.edu.cn.*

Manuscript received April 19, 2005; revised August 26, 2015.

This paper intends to explore the connection between local patterns and global shape for the point cloud completion task. To do so, we introduce a two-stage coarse-to-fine framework as shown in Figure 1. In the first stage, we learn a coarse skeleton containing global shape information, used as anchor points for consecutive detail enhancement. In the second stage, we establish the correlation between skeleton anchor points and local pattern features and enhance geometric details conditioned on the coarse skeleton.

Specifically, inspired by the Transformer Model [17] from natural language processing, we propose Skeleton-Detail Transformer, which contains a cross-attention layer and self-attention layers. The cross-attention layer is applied upon the partial input and coarse skeleton to effectively integrate local pattern features into coarse skeleton anchor points. The self-attention layer is used to better propagate the diverse information across the point set in a global view.

ECG [12] and VRCNet [18] also follows similar two-stage completion. However, they directly combine the coarse skeleton and partial input and treats them equally for following feature learning with local convolutional operations, which cannot effectively learn a globally explicit correlation between the coarse skeleton and local pattern.

Moreover, as the proposed attention layers are inspired by Transformer Model [17], it requires a quadratic dot-product computation, and $\mathcal{O}(N^2)$ memory usage on a N points point cloud, which is the major drawback in enhancing prediction capacity. Inspired by [19], we notice that only a few dot-product pairs contribute to the major attention; the others can be ignored safely. We thus propose a selective attention mechanism to only take the most likely points as query points for feature extraction, which can efficiently reduce memory usage without significant performance reduction.

To summarize, our main contributions are:

- We propose a coarse-to-fine point cloud completion network with a novel skeleton-detail transformer, exploring the correlation between local patterns and the generated coarse skeleton for more efficient detail recovery.
- We propose a selective attention mechanism that can greatly reduce memory usage without significantly affecting performance.
- Our proposed skeleton-detail transformer module can also be plugged into current completion networks.

2 Related Work

Non-learning based shape completion. Shape completion has long been a widespread problem of interest in the graphics and vision fields. Some effective descriptors have been developed in the early years, such as [20], [21], [22], which leverage geometric cues to fill the missing parts on the surface. However, these methods are usually limited to filling small holes. Another way for shape completion is to utilize a symmetry prior [23], [24], [25]. However, the assumption is too strong for general scenarios. Some researchers also proposed data-driven methods [26], [27], [28] which usually retrieve the most similar model based on the partial input from a large 3D shape database. While sometimes good results can be obtained, these methods are time-consuming in the matching process according to the database size.

Learning-based shape completion. Learning-based methods on shape completion usually use a deep neural network with an encoder-decoder architecture to directly map the partial input to a complete shape. Most pioneering works [1], [7], [8], [9], [10] rely on volumetric representations in which convolution operations can be applied directly. Since volumetric representations lead to large computation and memory costs, most works operate on low-dimension voxel grids, leading to detail missing. In contrast, PCN [11] directly generates complete shapes with partial point cloud as input by decoding a global latent feature. Following works [16], [29], [30], [31], [32], [33], [34], [35] improve the encoder-decoder architecture to recover more refined completion results. Other works [36], [37] tackles the point cloud completion in unpaired tasks. These approaches generate completed point clouds via decoding a global feature vector with very limited capability to represent geometric details.

More recent works [16], [32], [33], [34], [35], [38], [39], [40] have made efforts to preserve the observed geometric details from the local features in incomplete inputs. NSFA [15] separately reconstruct the unknown and known parts. VRC-Net [18] proposes a variational framework by leveraging the relationship between structures during the completion process. Pmp-net [16] accomplish the completion task by learning point moving paths moving paths. Snowflakenet [41] proposes a snowflake point deconvolution for point cloud completion. There are also some networks using voxel-based completion process. GRNet [14] proposes a gridding network for dense point reconstruction. SK-PCN [42] leverages attention mechanism to predict displacements for the skeleton and finally combine them with the input. VE-PCN [43] develops a voxel-based network for point cloud completion by leveraging edge generation. PoinTR [38] uses a geometry transformer to predict the missing shape.

Both SK-PCN [42] and our approach use an attention mechanism to obtain refined results. The main difference is how SK-PCN and our method get the refined results from the skeleton. SK-PCN combines input and skeleton features through a non-local attention module to predict displacements for each skeletal point, and add these displacements to the skeleton to get the final results. While in our method, we leverage a skeleton-detail transformer to enhance the skeleton feature, and directly reconstruct the entire shape from the enhanced features. PoinTR [38] leverages transformer architecture for point completion, but its formulation is quite different from ours. The main difference is that PoinTR designed a geometry-aware transformer to predict the missing part, formulating the point cloud completion task as a set-to-set translation task. Our method follows a coarse-to-fine pipeline and designs a skeleton-detail transformer aiming to enhance the skeleton details and directly predict the entire shape.

Transformer on point cloud. Transformer and self-attention models have revolutionized machine translation and natural language processing [17], [44], [45], [46], [47]. This has inspired the development of self-attention networks for 2D image recognition [48], [49], [50], [51]. Recently some works are proposed for processing point cloud with transformer models. [52], [53], [54] try to design general point cloud transformer on 3D processing task such as classification and segmentation. [55] proposes a transformer model on 3D object detection. [56] propose a pyramid point cloud transformer for large-scale place recognition.

3 Network Architecture

In this section, we elaborate on the detail of our coarse-to-fine point completion framework with an overview shown in Figure 2.

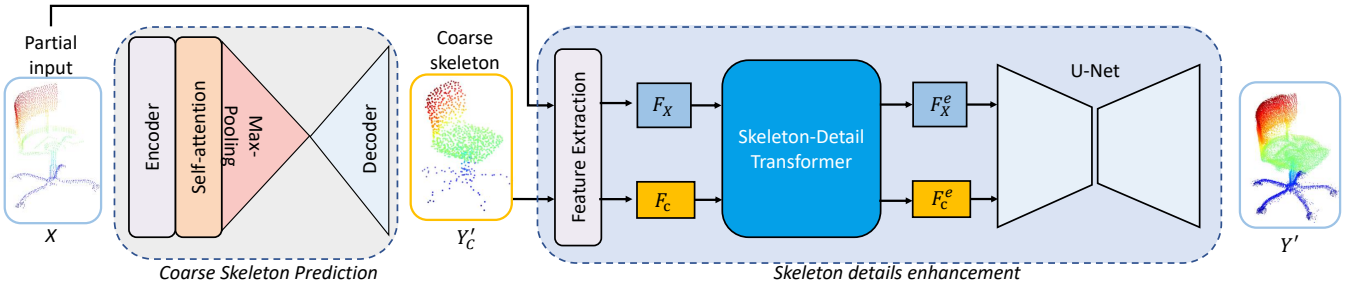


Fig. 2: Overall network architecture. It consists of two stages, which respectively learn the coarse skeleton, and further enhance the skeleton with local details by a skeleton-detail transformer.

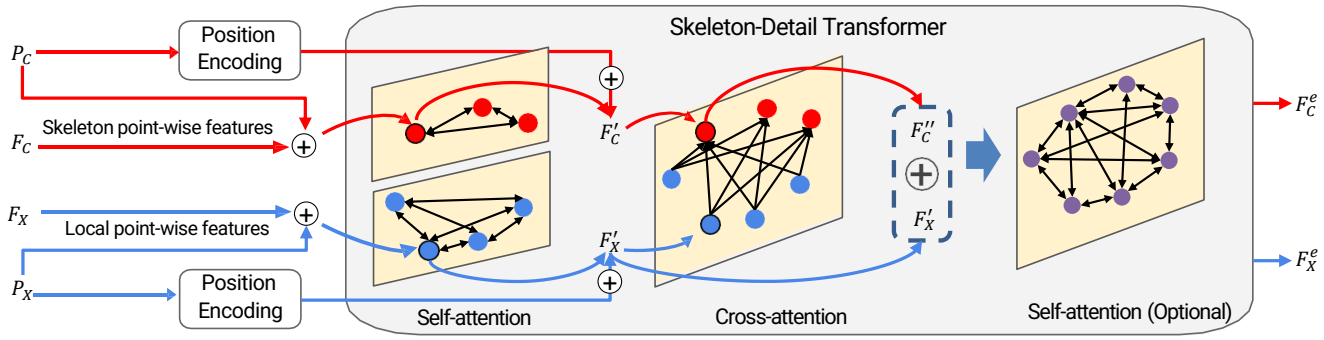


Fig. 3: Skeleton-Detail Transformer architecture. It consists of a self-attention layer, a cross attention layer, and an optional globally self-attention layer. The self-attention layer is performed respectively in query and key feature sets, and the cross attention layer integrates features from local patterns to features of skeleton anchor points. An optional global self-attention layer can be applied to the combination of enhanced features F'_X and F''_C to further explore the correlations in a global view.

Given X as a partial input point cloud, we first leverage a PCN auto-encoder [11] to generate the global feature for coarse shape completion. A coarse skeleton Y'_C which refers to the coarse, intermediate completion shape is obtained by decoding the global feature. However, different from PCN, we additionally involve a self-attention layer (Section 4.2) before the max-pooling layer to better aggregate local features. Subsequently, we use an MLP to extract point-wise features F_X and F_C from X and Y'_C, respectively. F_X can be seen as local features containing geometric details, and F_C refers to the global skeleton points features. We then feed F_X and F_C to a Skeleton-Detail Transformer (Section 4) to integrate the local patterns features from F_X to F_C and get enhanced features F_X^e and F_C^e. Finally, the combination of F_X^e and F_C^e are fed into a reconstruction network with U-Net architecture to get the final details and enhanced results.

ECG [12] also follows a similar two-stage completion, but in the second stage, it directly combines Y'_C with X and treat them equally in the following process with local feature aggregation operations for detail refinement.

Different from ECG, to address the correlation from the local pattern to the overall skeleton, we feed F_X and F_C to our skeleton-detail transformer, which is detailed in Section 4, to effectively integrate the local pattern features from F_X to F_C in a global view.

4 Skeleton-Detail Transformer

Our skeleton-detail transformer, as illustrated in Figure 3, consists of a self-attention layer, a cross attention layer, and an optional global self-attention layer. The inputs are F_X and F_C, the point-wise features of X and Y'_C. P_X and P_C are the corresponding positions. The self-attention layer aggregates feature in each point set with output F'_X and F'_C. The cross attention layer elaborates on exploring the correlation and integrates features F'_X from local patterns to features F'_C of skeleton points, where we get the enhanced feature F''_C. Finally, an optional global self-attention layer can be applied to the combination of F'_X and F''_C to propagate the features further in a global view. The global self-attention layer can boost the performance in our experiments. However, it requires extra computation and memory usage accordingly, so we consider it optional and discuss it in Section 8.7.

We first briefly introduce the general formulation of the Transformer Model [17]. Then we present the detail of our self-attention layer and cross-attention layer. Finally, we introduce a selective attention mechanism.

4.1 Background of Transformer Model

Let $\mathcal{F} = \{f_i\}$ be a set of feature vectors and $\mathcal{P} = \{p_i\}$ be the corresponding positions. The standard self-attention layer with a single-attention head in Transformer Model [17] first computes a tuple (query, key, value) and performs the scaled dot-product as:

$$q_i = f_i W_q, k_i = f_i W_k, v_i = f_i W_v, \tag{1}$$

$$y_i = \sum_{p_i \in \mathcal{P}} \sigma(q_i k_j^T / \sqrt{d} + \mathrm{PE}(p_i, p_j)) v_j, \tag{2}$$

where W_q, W_k, W_v are weight matrices for query, key, and value. d is the feature dimension of f_i. $\mathrm{PE}(\cdot)$ is the positional encoding function for input positions. $\sigma(\cdot)$ is a normalization function, and softmax is mostly adopted. There also can be multi-attention heads when computing (q, k, v), but we use single-head attention considering the memory and computation efficiency.

4.2 Self-attention Layer

We first introduce a general point transformer, a self-attention layer for learning feature representations inside a point set. Formally, given an input point cloud $\mathcal{P}$ with embedded feature $\mathcal{F}$, we formulate the self-attention layer as follows:

$$q_i = \alpha(f_i), k_i = \beta(f_i), v_i = \gamma(f_i), \tag{3}$$

$$y_i' = \sum_{f_j \in \mathcal{F}} \sigma(q_i k_j^T / \sqrt{d}))v_j, \tag{4}$$

$$y_i = f_i + \delta(f_i - y_i'), \tag{5}$$

where α, β, γ are linear projections, and σ is an MLP with one linear layer. δ is an MLP containing one linear layer with batch normalization and ReLU nonlinearity. We add Eq. 5 as [52] indicates that calculating the offset between the self-attention features and the input features can get better feature representations. Positional embedding is discarded in the self-attention layer since the point features obtained from the point coordinates $\mathcal{P}$ already contain enough positional information.

4.3 Cross-attention Layer

We note that directly applying a self-attention layer on the combination of F_X and F_C does not get good performance. Since the coarse skeleton is recovered from a single global vector, limited information is shared between F_X and F_C. To this end, we propose a cross-attention layer to fully explore the correlation from local patterns to skeleton anchor points before the combined feature aggregation. The formulation is similar to our self-attention layer, but with two important modifications: 1) In the cross-attention layer, only F_C' are used for computing the queries, and the keys and values are computed from F_X', indicating that we learn a cross-mapping $F_X' \rightarrow F_C'$. 2) An additional positional encoding layer is involved. We found that adding a position encoding layer can significantly boost the performance for finding long-range relations from the local patterns to the skeleton anchor points.

[57], [58] show that mapping the input to a higher dimensional space using high-frequency functions before passing them to the network enables a better fitting of data that contains high-frequency variation. Inspired by [58], given the position p, we define a mapping function γ which is a mapping from $\mathbb{R}^L$ into a higher dimensional space $\mathbb{R}^{2L}$. Formally, the encoding function we use is:

$$\mathrm{PE}(p_i, p_j) = MLP(\gamma(p_i) - \gamma(p_j)), \tag{6}$$

$$\begin{aligned}\gamma(p) = (&\sin(2^0\pi p), \cos(2^0\pi p), \cdots, \\ &\sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))\,,\end{aligned} \tag{7}$$

where we set $L = 3$ in our architecture.

Give the point-wise features $f_c \in F_C$, $f_x \in F_X$, we formulate the cross-attention layer as follows:

$$q_i = \alpha(f_{c_i}), k_i = \beta(f_{x_i}), v_i = \gamma(f_{x_i}), \tag{8}$$

$$y_i' = \sum_{p_i \in Y_C', p_j \in X}^{j} \sigma(q_i k_j^T / \sqrt{d} + \mathrm{PE}(p_i, p_j))v_j, \tag{9}$$

$$y_i = f_{c_i} + \delta(f_{c_i} - y_i'), \tag{10}$$

where $\alpha, \beta, \gamma, \sigma$ and δ have the same meaning with self-attention layer.

Algorithm 1 Selective attention mechanism

Input: tensor $\mathbf{Q} = \{q_i\} \in \mathbb{R}^{m \times d}, \mathbf{K} = \{k_i\} \in \mathbb{R}^{n \times d}$, $\mathbf{V} = \{v_i\} \in \mathbb{R}^{n \times d}$. Corresponding features $\mathbf{F_q}$ of $\mathbf{Q}$
Initialize: set sampling factor s, $U = s \times m$, $\mathbf{A} = \mathbf{F_q}$

1: feed $\mathbf{Q}$ to an MLP with two linear layers to get a selective map $\mathbf{M} \in \mathbb{R}^{m \times 1}$
2: select top U queries under $\mathbf{M}$ as $\overline{\mathbf{Q}}$
3: set $\mathbf{A}^* = \sigma(\overline{\mathbf{Q}}\mathbf{K}^\mathbf{T}/\sqrt{d})\mathbf{V}$
4: replace the top U items in $\mathbf{A}$ with $\mathbf{A}^*$ by their original rows accordingly:

Output: attention feature map $\mathbf{A}$

While this position encoding can also be used in our self-attention layer, we note that the performance improvement is insignificant. We give an intuitive explanation: in the cross-attention layer, the mutual information between F_C' and F_X' is very limited, as the coarse skeleton is recovered from a single global vector, making it hard to establish the correlation without more specific positional information. While in the following global self-attention layer, since a correlation has already been established, it is easier for feature learning even without additional position information. Also, it is not hard to find the relations in the same point set in the first self-attention layer by only leveraging the point coordinates.

Moreover, computing the relative position encoding in every attention layer is memory-consuming, requiring $\mathcal{O}(N_q \times N_k)$ memory usage and extra computation (N_q and N_k are the point numbers in query and key).

4.4 Selective Attention mechanism

Our designed attention layers require quadratic times dot-product computation and $\mathcal{O}(N_q \times N_k)$ memory usage, which causes heavy resource consumption. A recent work [19] observes that the attention map after the dot-product is potentially sparse in several NLP tasks, which means only a few dot-product pairs contribute to the major attention, and others can be ignored. We find this observation is also consistent in our completion task.

To this end, we design a selective attention mechanism that selects the most 'important' query points to perform the attention operation, rather than considering all the points. We give a formulation of our selective attention mechanism in Algorithm 1 and an illustration in Figure 5. Specifically, we use an MLP with two linear layers, which extracts each point-wise feature to a scalar as a selective map. We then select the top U queries to compute new features $A*$ and update the corresponding U items in A by the computed features $A*$. The proposed selective attention mechanism can be applied in both the self-attention and cross-attention layers to replace Eq. 4 and Eq. 9, where the unselected queries will finally have the original feature f_i without updates.

We find our selective attention mechanism could save the memory usage without significantly affecting performance, which is discussed in Section 8.7.

Intuitively, we can also turn to select the most 'important' keys when aggregating the features to a specific query, but different queries will require different score maps for each key. It will need $\mathcal{O}(N_q \times N_k)$ memory for the score maps, but our mechanism only needs $\mathcal{O}(N_q)$ memory.

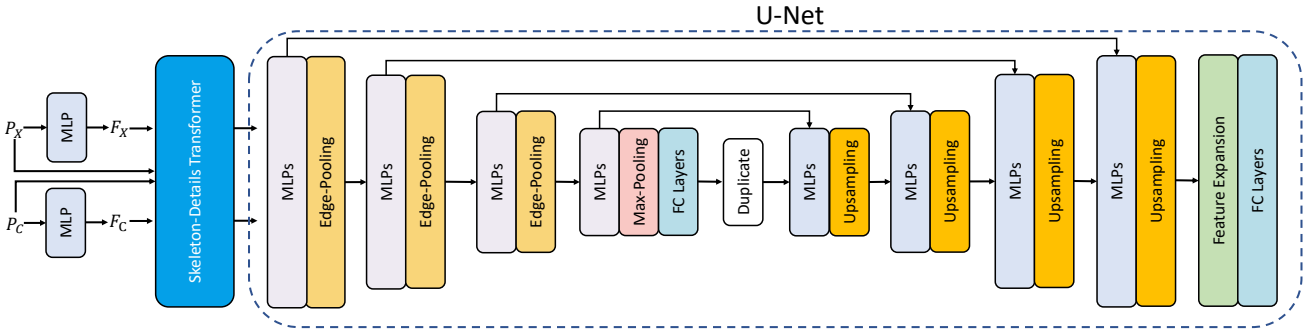


Fig. 4: Reconstruction network. It use a U-Net architecture composed of MLPs, downsampling and upsampling layers, FC layers, and an expansion module.

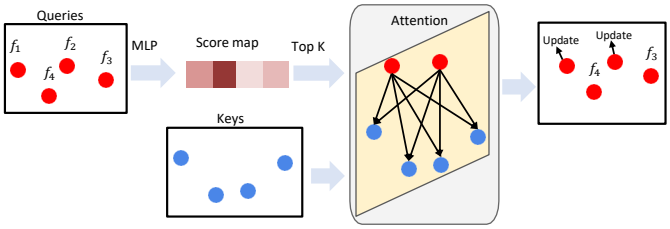


Fig. 5: Illustration of selective attention mechanism.

5 Reconstruction Network

After the skeleton-detail transformer, we get the enhanced features F_X^e and F_C^e from F_X and F_C. We then combine them as the input to the reconstruction network with the architecture shown in Figure 4.

The reconstruction network follows a U-Net architecture with skip connections. We use MLP as the basic module for feature propagation. For each downsampling and upsampling operation, we leverage Edge-preserved Pooling and Edge-preserved Unpooling modules in Pointatrousgraph [12]. Also, an Edge-aware Feature Expansion (EFE) module [59] is used to expand point features depending on the required completion resolution. The parameter detail of the network is illustrated in the supplementary material.

6 Loss Function

Chamfer Distance (CD) and Earth Mover's Distance (EMD) are introduced [60] to measure the differences between two point clouds (P, Q). We choose the Chamfer distance due to its efficiency over EMD:

$$\mathcal{L}_{CD}(P,Q) = \frac{1}{|P|}\sum_{x\in P}\min_{y\in Q}\|x-y\|_2 + \frac{1}{|Q|}\sum_{y\in Q}\min_{x\in P}\|y-x\|_2. \tag{11}$$

Different from previous works [12], [15] which add additional uniform loss or repulsion loss to the completion results, we only leverage Chamfer Distance on the coarse skeleton and final result. We jointly train the network by minimizing the loss as:

$$\mathcal{L} = \alpha\mathcal{L}_{CD}(Y'_C, Y_{gt}) + \mathcal{L}_{CD}(Y', Y_{gt}). \tag{12}$$

7 Implementation Details

We use a Pytorch implementation for our model, trained for 100 epochs with a batch size of 32 and an Adam optimizer. The initial learning rate is set to 0.0001, decaying by 0.7 for every 20 epochs. The point number in the generated coarse skeleton is set to 1024.

Self-attention layers Given the following definition:

$$\begin{aligned} q_i = \alpha(f_i), k_i = \beta(f_i), v_i = \gamma(f_i),\\ y'_i = \sum_{p_i\in\mathcal{P}}^{p_j} \sigma(q_i k_j^T/\sqrt{d} + \mathrm{PE}(p_i,p_j))v_j,\\ y_i = f_i + \delta(f_i - y'_i),\\ \mathrm{PE}(p_i,p_j) = MLP(\gamma(p_i) - \gamma(p_j)),\\ f_i \in \mathbb{R}^{d_f}, q_i, k_i \in \mathbb{R}^{d_{qk}}, v_i \in \mathbb{R}^{d_v}. \end{aligned} \tag{13}$$

We set $d_{qk} = d_f/4$ and $d_v = d_f$. For PE$(\cdot)$, we use the same implementation in [12] with $L = 3$, and the MLP in PE$(\cdot)$ restored the features from $2L$ channels to 1 channel.

Coarse completion auto-encoder We use the auto-encoder in PCN with the same parameters, but apply a self-attention layer to the point-wise features before the max-pooling layer.

U-Net Figure 6 details our U-Net specification for detail enhancement. We add skip-connections within each network hierarchy similar to MPU [66]. MLP parameters are given as (input_channel, output_channel).

8 Experiments

We test our method on two synthetic datasets and two real scanned datasets. The model with the optional global self-attention layer is used as our method if without a special description.

8.1 Datasets

PCN PCN [11] creates a dataset based on a subset of the Shapenet [67] dataset. In our experiments, complete point clouds contain 16384 points and 2048 points for partial point clouds. The training set includes 28974 different models from 8 categories. Each model has a complete point cloud with 8 partial point clouds taken from different viewpoints for data augmentation. The validation set contains 100 models. The testing contains 1200 models with 150 models in 8 categories.

Completion3D The Completion3D benchmark [29] is composed of 28,974 and 800 samples from ShapeNet dataset for training and validation, respectively. Unlike the ShapeNet dataset generated by PCN, there are only 2,048 points in the ground truth point clouds.

Method	Avg.		Plane		Cabinet		Car		Chair		Lamp		Couch		Table		Watercraft	
	CD	F1	CD	F1	CD	F1	CD	F1	CD	F1	CD	F1	CD	F1	CD	F1	CD	F1
AtlasNet [61]	10.85	0.616	6.37	0.845	11.94	0.552	10.10	0.630	12.06	0.552	12.37	0.565	12.99	0.500	10.33	0.660	10.61	0.624
FoldingNet [62]	14.31	0.299	9.49	0.322	15.80	0.642	12.61	0.237	15.55	0.382	16.41	0.236	15.97	0.219	13.65	0.197	14.99	0.361
PCN [11]	9.64	0.695	5.50	0.881	22.70	0.651	10.63	0.725	**8.70**	0.625	11.00	0.638	11.34	0.581	11.68	0.765	8.59	0.697
TopNet [29]	12.15	0.503	7.61	0.771	13.31	0.404	10.90	0.544	13.82	0.413	14.44	0.408	14.78	0.350	11.22	0.572	11.12	0.560
MSN [33]	-	0.705	-	0.885	-	0.644	-	0.665	-	0.657	-	0.699	-	0.604	-	0.782	-	0.708
GRNet [14]	8.83	0.708	6.45	0.843	10.37	0.618	9.45	0.682	9.41	0.673	7.96	0.761	10.51	0.605	8.44	0.751	8.04	0.750
Wang et al. [13]	8.51	0.652	4.79	0.918	**9.97**	0.379	8.31	0.687	9.49	0.637	8.94	0.603	10.69	0.517	7.81	0.721	8.05	0.759
PMP-Net [16]	8.73	-	5.65	-	11.24	-	9.64	-	9.51	-	**6.95**	-	10.83	-	8.72	-	**7.25**	-
ECG [12]	8.63	0.724	5.23	0.899	10.12	0.631	8.36	0.704	9.43	0.687	8.53	0.755	10.94	0.579	7.98	0.790	8.16	0.750
NSFA [15]	8.32	0.734	5.03	0.896	10.51	0.629	9.11	0.674	9.16	0.686	7.45	0.793	**10.46**	0.608	**7.56**	0.806	7.28	0.781
SK-PCN [43]	8.49	0.736	5.09	0.911	9.98	0.643	8.22	0.716	9.29	0.699	8.39	0.767	10.80	0.591	7.84	0.802	8.02	0.762
Pointr [38]	8.38	**0.754**	4.75	0.915	10.47	**0.665**	8.68	0.718	9.39	0.710	7.75	**0.798**	10.93	**0.632**	7.78	0.796	7.29	**0.797**
Ours	**8.24**	**0.754**	**4.60**	**0.924**	10.05	0.659	**8.16**	**0.733**	9.15	**0.724**	8.12	0.795	10.65	0.609	7.64	**0.807**	7.66	0.778

TABLE 1: Quantitative comparisons on PCN dataset with state-of-the-art methods in terms of L1 Chamfer Distance $\times 10^{-3}$ and F1-Score.

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
FoldingNet [62]	19.07	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51
PCN [11]	18.22	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73
PointSetVoting [63]	18.18	6.88	21.18	15.78	22.54	18.78	28.39	19.96	11.16
AtlasNet [61]	17.77	10.36	23.40	13.40	24.16	20.24	20.82	17.52	11.62
SoftPoolNet [64]	16.15	5.81	24.53	11.35	23.63	18.54	20.34	16.89	7.14
TopNet [29]	14.25	7.32	18.77	12.88	19.82	14.60	16.29	14.89	8.82
SA-Net [16]	11.22	5.27	14.45	7.78	13.67	13.53	14.22	11.75	8.84
GRNet [14]	10.64	6.13	16.90	8.27	12.23	10.22	14.93	10.08	5.86
PMP-Net [16]	9.23	3.99	14.70	8.55	10.21	9.27	12.43	8.51	5.77
Wang et al. [13]	9.21	3.38	13.17	8.31	10.62	10.00	12.86	9.16	5.80
SCRN [65]	9.13	3.35	12.81	7.78	9.88	10.12	12.95	9.77	6.10
VRCNet [18]	8.12	3.94	10.93	6.44	9.32	8.32	11.35	8.60	5.78
VE-PCN [43]	8.10	3.83	12.74	7.86	8.66	**7.24**	11.47	7.88	**4.75**
SnowflakeNet [41]	**7.60**	3.48	11.09	6.90	8.75	8.42	**10.15**	**6.46**	5.32
Ours	7.78	**2.74**	**10.25**	**6.33**	**8.4**	9.85	11.06	7.54	5.88

TABLE 2: Quantitative comparisons on Completion3D dataset with state-of-the-art methods in terms of L2 Chamfer distance $\times 10^{-4}$.

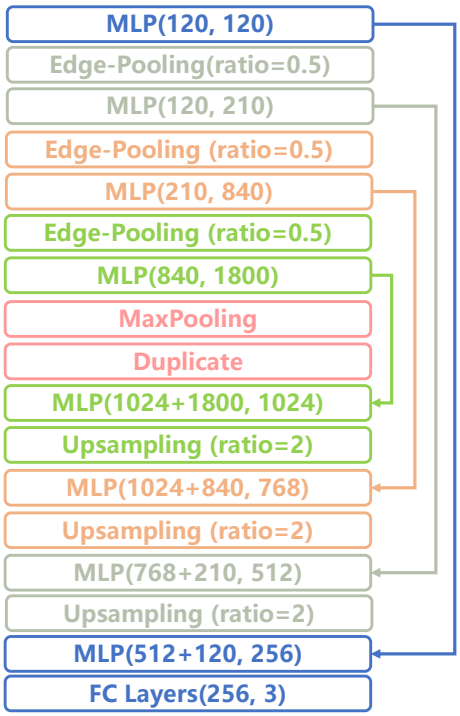


Fig. 6: Detailed U-Net parameters.

Kitti We also test our methods on real-world scans on Kitti [68]. The testing scans are cars extracted from each frame according to the ground truth object bounding boxes. The testing set contains 2483 partial point clouds labeled as cars.

ScanNet The ScanNet [69] datasets are obtained from [36], which extracts 550 chairs objects and 550 tables from ScanNet dataset, and manually aligns each model to be consistently orientated with models in ShapeNet dataset. We use the trained model on PCN dataset for testing on both real scanned datasets.

8.2 Evaluation Metrics

Besides the Chamfer Distance introduced in Section 6, we also use the F1-Score metric introduced in [14]. Let $\mathcal{T} = \{(x_i, y_i, z_i)\}_{i=1}^{n_\mathcal{T}}$ be the ground truth and $\mathcal{R} = \{(x_i, y_i, z_i)\}_{i=1}^{n_\mathcal{R}}$ be a reconstructed point set being evaluated, where $n_\mathcal{T}$ and $n_\mathcal{R}$ are the numbers of points of $\mathcal{T}$ and $\mathcal{R}$ respectively. F1-Score is defined as:

$$\text{F1-score}(d) = \frac{2P(d)R(d)}{P(d) + R(d)}, \tag{14}$$

where $P(d)$ and $R(d)$ denote the precision and recall for a distance threshold d, respectively.

$$P(d) = \frac{1}{n_\mathcal{R}} \sum_{r \in \mathcal{R}} \left[\min_{t \in \mathcal{T}} ||t - r|| < d \right], \tag{15}$$

$$R(d) = \frac{1}{n_\mathcal{T}} \sum_{t \in \mathcal{T}} \left[\min_{r \in \mathcal{R}} ||t - r|| < d \right]. \tag{16}$$

Unlike Chamfer distance, a higher F1-Score means better performance.

8.3 Completion Results on PCN dataset

We also compare our network on PCN dataset with several state-of-the-art baseline methods. L1 Chamfer Distance is involved as the evaluation metric. In addition, as is pointed out in [70] that CD can be misleading in some cases, F1-Score [71] is also used for evaluation in our experiments. Compared with CD, which computes the distance between two point clouds, F1-Score can better judge the similarity between two surfaces. For the baseline methods, the results of [12], [15] are produced from the codes and pre-trained models released in their official projects at Github, and the other methods are cited from [14], [16] and their original papers. We set the sampling ratio to 1.0 in the selective attention mechanism to get the best performance.

The quantitative results are shown in Table 1. It is notable that our network achieves the highest F1-Score and lowest L1 CD on average. Also, it is notable that though our method performance is not as good as NSFA on CD in some categories such as chair and lamp, our method gets a better F1-Score than NSFA, indicating our method can guarantee a more accurate surface. This can also be observed in the visual comparison. PoinTR achieves a similar F1-Score with our method, but our method achieves a lower L1 CD.

Figure 7 shows the qualitative comparison of our method. While other methods can preserve the detail in completed results, the distortion is also visible. In contrast, our results predict a more accurate shape with detailed patterns and less distortion. For instance, in Figure 7 (a), our method can better recover the missing leg of the chair while preserving the details. In contrast, results from other methods are very noisy, which demonstrates the ability of our network to enhance the shape with local details efficiently. More visual comparisons are shown in Figure 11.

We also compare our method with methods that support multi-resolution completion with different output point numbers in Table 3. It shows that our network outperforms other baselines in various resolutions.

Points	2048		4096		8192	
	CD	F1	CD	F1	CD	F1
PCN [11]	14.17	0.417	12.30	0.554	10.77	0.649
Wang et al. [13]	15.28	0.387	13.43	0.510	11.85	0.600
ECG [12]	13.01	0.462	13.43	0.510	9.40	0.701
GRNet [14]	12.86	0.384	11.14	0.534	9.78	0.648
NSFA [15]	12.54	0.485	10.69	0.630	9.16	0.724
Ours	**12.46**	**0.490**	**10.68**	**0.636**	**9.14**	**0.725**

TABLE 3: Results on PCN dataset with multi-resolution.

8.4 Completion Results on Completion3D

Following GRNet [14], we adopt the model with the lowest CD on the validation set and recover point clouds on the Completion3D testing set. Next, random subsampling is applied to the generated point clouds to obtain 2,048 points for benchmark evaluation. The results are reported in Table 2 in accordance with the completion3D leaderboard. Our network ranks first on the Completion3D benchmark.

8.5 Completion Results on Kitti / ScanNet

As there is no complete ground truth for Kitti and ScanNet, we therefore use two metrics proposed in PCN to quantitatively evaluate the performance: 1) Fidelity (Fid.) error, which is the average distance from each point in the input to its nearest neighbor in the output. This measures how well the input is preserved. 2) Minimal Matching Distance (MMD), which is the L1 Chamfer Distance between the output and the car/table/chair point cloud from ShapeNet that is closest to the output point cloud in terms of CD. This measures how much the output resembles a typical car/table/chair.

Methods	Kitti		ScanNet Chair		ScanNet Table	
	Fid.	MMD	Fid.	MMD	Fid.	MMD
PCN [11]	1.73	15.75	13.38	22.79	8.36	29.94
GRNet [14]	1.94	27.65	5.92	21.66	5.11	18.37
NSFA [15]	1.03	19.85	12.24	34.55	8.72	28.48
SK-PCN [42]	0.21	16.36	**3.14**	22.30	**2.21**	16.14
Pointr [38]	**0.00**	15.31	-	-	-	-
Ours	0.96	**14.36**	11.46	**16.74**	7.67	**13.77**

TABLE 4: Quantitative comparisons on Kitti and ScanNet datasets with Fidelity error and MMD $\times 10^{-3}$.

The quantitative and qualitative results are shown in Table 4 and Figure 8,9 respectively. Our method achieves the lowest MMD and more plausible visualization results in both results. GRNet [14] and SK-PCN [42] achieve the lower fidelity error in table and chair, but the MMD is higher than ours. The qualitative comparison shows that other methods also present reasonable results, but the distortion and blur are obvious. The fidelity error in Kitti dataset is significantly lower than ScanNet due to the fewer point number of partial input as shown in Figure 8. PoinTR [38] achieves zero fidelity error as it totally preserve the input points in the final results.

8.6 Visualization of how the cross-transformer aggregates local features

As we have claimed the ability of our proposed cross-attention module to aggregate local features into the global skeleton, we wonder how the cross-attention aggregates local features on different parts of the global skeleton. To this end, we visualize a heatmap of the local input features for different query points in the coarse skeleton. We visualize the weight w_{ij} of each point p_j in partial input to the certain query point p_i in coarse skeleton. Specifically, according to Equation 9, w_{ij} is defined as:

$$w_{ij} = \sigma(q_i k_j^T / \sqrt{d} + \mathrm{PE}(p_i, p_j)). \tag{17}$$

Note that the weights are normalized to [0,1].

Visualization results are shown in Figure 10. The visualization shows that the transformer module can aggregate long-distance information based on the relationship for a certain query point. For example, the transformer module tends to aggregate the corresponding wheel information for a point near the missing wheel in Figure 10 (I)(a). In Figure 10 (II)(a), high weights are given to the chair legs in partial input to provide more cues to complete the missing leg of the chair. In addition, we can see that the details are enhanced after the corresponding features are aggregated to the coarse skeleton features through our cross-transformer.

Another interesting observation is that in Figure 10 (III)(c), the transformer gives high weights for both the propeller and the wing for a point in the head of the plane. We consider this because the shape of the incomplete wing is similar to the plane's propeller.

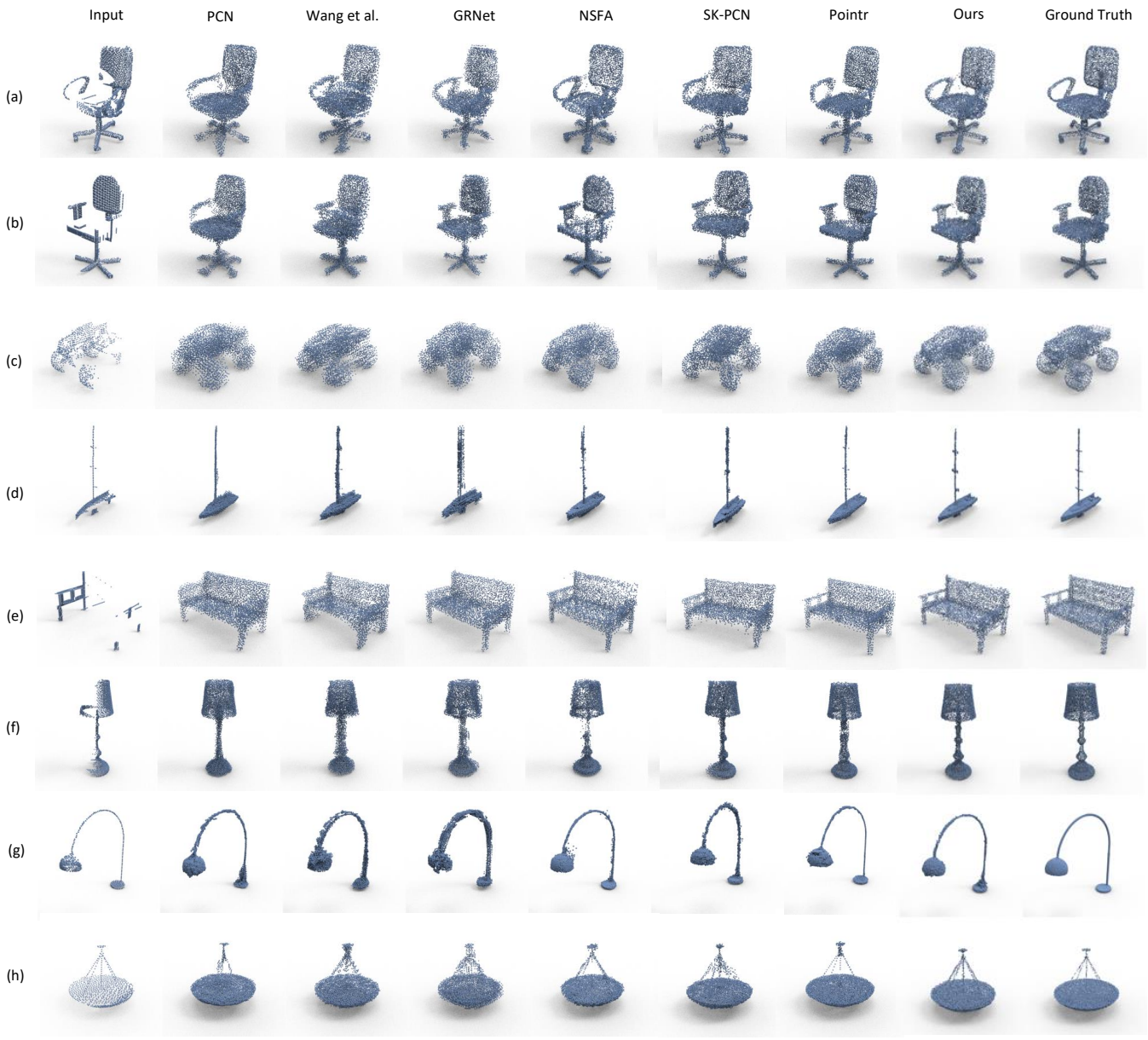


Fig. 7: Visual comparisons with state-of-the-art methods on ShapeNet dataset.

1st stage	2nd stage			CD		F1		Memory	Model	Inf.
SA	SA	CA	GSA	coarse	completion	coarse	completion	Usage(Mib)	Size(Mib)	Time(ms)
				15.40	8.69	0.297	0.724	348.86	54.18	213
✓				15.21	8.54	0.301	0.730	424.85	56.69	236
✓			✓	15.23	8.43	0.303	0.736	512.32	56.83	251
✓	✓	✓		**14.91**	8.31	0.307	0.746	518.65	57.11	260
✓		✓	✓	15.01	8.27	0.307	0.750	550.65	56.97	258
✓	✓	✓	✓	**14.91**	**8.24**	**0.313**	**0.753**	606.12	57.25	295

TABLE 5: Ablation study on different combinations of attention layers.

8.7 Ablation studies

We give ablation studies on our network with the output of 16384 points and batch size 1.

Evaluation of selective attention mechanism. We give a study of our selective attention mechanism about the memory usage in the forward/backward pass, the layer size, and the inference (Inf.) time of a single attention layer with different sampling factors in Table 6, where the point number of the coarse skeleton is 1024. The CD and F1-Score on final completion results are also reported with the selective attention mechanism applied to all attention layers of the network. Table 6 also shows that the network can produce reasonable results with a sampling factor bigger than 0.5.

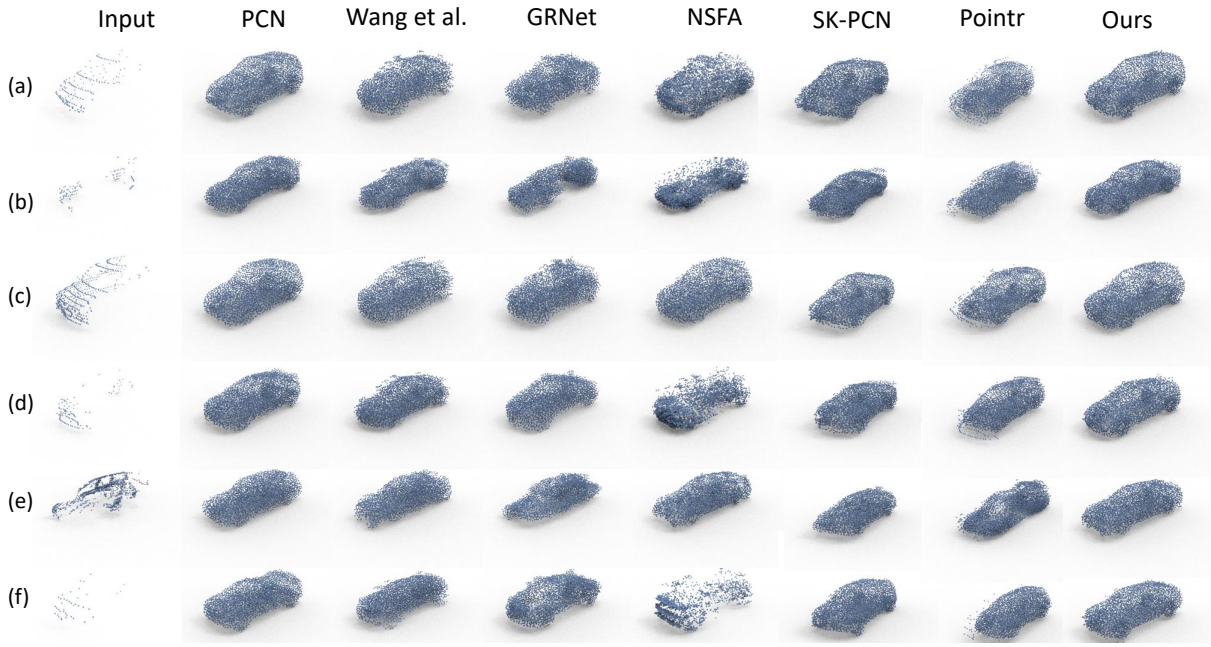


Fig. 8: Visual comparisons with state-of-the-art methods on Kitti dataset.

Sampling factor	Memory Usage (Mib)	Layer. Size (Mib)	Inf. Time (ms)	CD	F1
0.25	17.98	0.19	28	9.23	0.691
0.5	26.09	0.19	29	8.47	0.716
0.75	34.21	0.19	31	8.34	0.739
1.0	42.33	0.19	32	8.24	0.753

TABLE 6: Efficiency of selective attention mechanism with different sampling factors.

Furthermore, we find that the performance of our selective attention mechanism is largely influenced by the point number of the coarse skeleton when performing cross-attention. For example, in Table 6, setting the factor to 0.25 will result in only 256 query points of the coarse skeleton to perform cross-attention, but 256 points are hard to cover the object surface. So we give further studies with 2048 and 4096 points in coarse skeleton in Table 7. It shows that the network can achieve better performance with a small sampling factor when the point number of the coarse skeleton becomes larger.

Sampling factor	1024 points	2048 points	4096 points
0.25	9.23	8.53	8.39
0.50	8.47	8.41	8.38
0.75	8.34	8.27	8.29
1.0	8.24	8.21	8.24

TABLE 7: Chamfer distance with different coarse skeleton point numbers on PCN dataset.

We also tried using FPS sampling when we initially designed our network. We downsample the point cloud and update the features of the points selected by FPS only. We show the results in Table 8. The original point number of the skeleton is 1024. We can observe that though the performance of using FPS is acceptable, our selective attention mechanism can achieve better results.

Evaluation of proposed attention layers. Table 5 demonstrates the ablation studies conducted on our proposed attention layers. In the first stage, we test the effectiveness of adding a self-attention layer. In the second stage, we test different combinations of self-attention, cross-attention, and global self-attention layers. CD and F1-Score from the first and second stages are reported, as well as the memory usage in the forward/backward pass, inference time, and model size.

Sampling factor	FPS	elective attention mechanism
0.25	11.21	9.23
0.50	9.32	8.47
0.75	8.39	8.34
1.0	8.24	8.24

TABLE 8: Chamfer distance with different points and selecting strategy on PCN dataset.

U-Net analysis. We also give an ablation study for the U-Net design. We try to replace the current U-Net with PointNet, PointNet++, DGCNN auto-encoder with the results shown in Table 9. We find that our U-Net achieves the best performance, and the results of DGCNN are also acceptable, but using vanilla PointNet, PointNet++ can hardly achieve acceptable results.

U-Net	CD	F1
PointNet	9.65	0.672
PointNet++	9.21	0.693
DGCNN	8.95	0.710
PointNet++ with skip connection	8.65	0.730
Ours	**8.24**	**0.754**

TABLE 9: Chamfer distance and F1-Score with different U-Net choice on PCN dataset.

8.8 Comparison about model resources usage

We report the resource usage by GRNet, NSFA, and our network in Table 10. GRNet and our network are implemented using PyTorch, and we use the official implementation (by TensorFlow) for NSFA. To achieve a fair comparison for the inference time, we use the same batch size 1 and test all methods using a single NVIDIA GTX 2080Ti on the same workstation. Our network

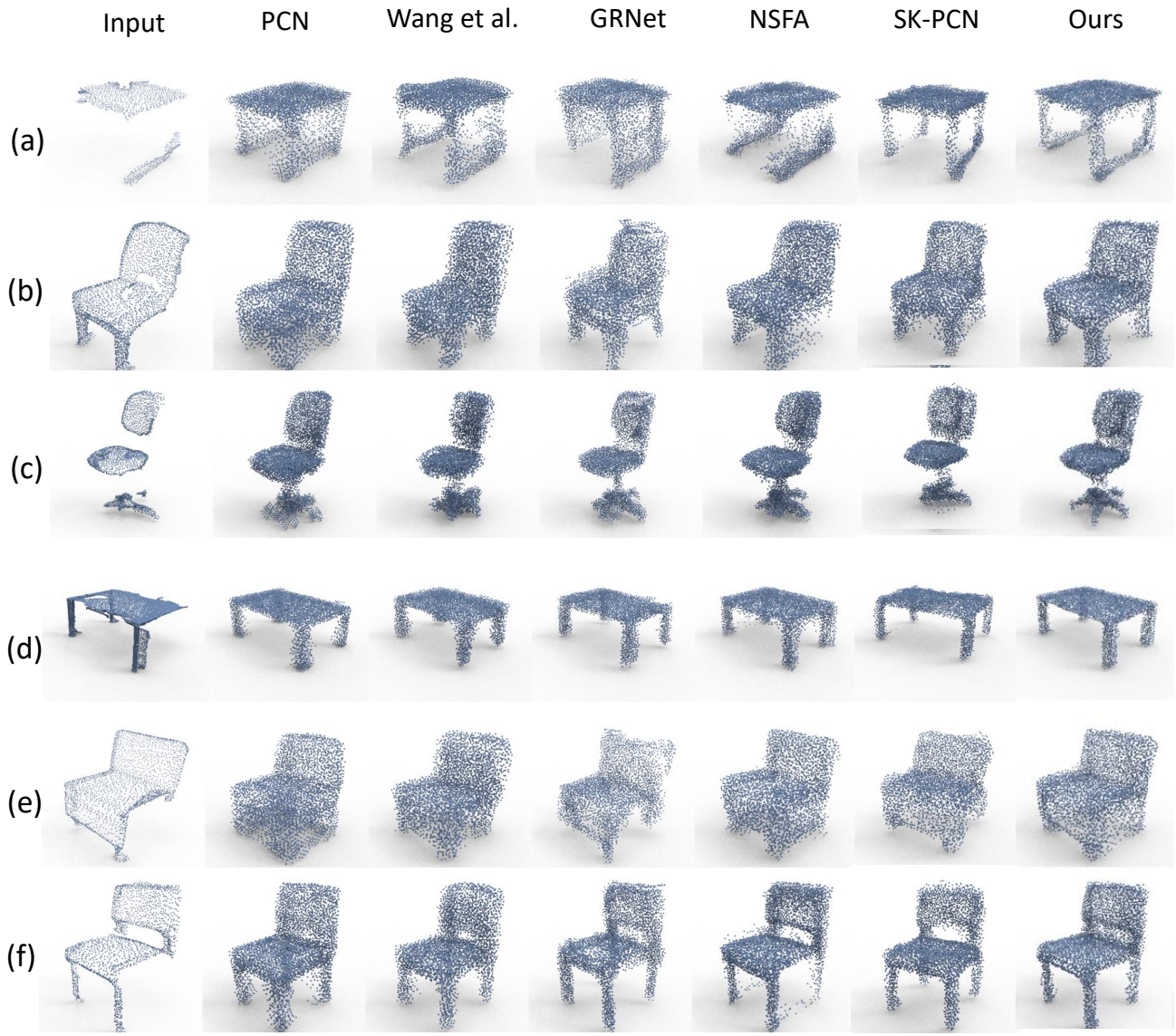


Fig. 9: Visual comparisons with state-of-the-art methods on ScanNet dataset.

has the least model size and achieves plausible improvements in completion qualities with an acceptable increment in the inference time.

Methods	Model size (Mib)	Inference time (ms)
GRNet	76.70	147
NSFA	64.01	234
Ours	57.11	260

TABLE 10: Model resources usage of different methods.

9 Discussion

Can the Skeleton-Detail Transformer be applied to other completion networks?

Our proposed transformer module is proposed to aggregate the local features from the partial input into the coarse completion shape. Theoretically, this transformer module can be applied to any completion network with a coarse-to-fine style.

To this end, we try to integrate our skeleton-detail transformer into other coarse-to-fine frameworks (Wang et al., VRCNet). Table 11 shows the results of these networks before and after being integrated with our transformer. Note that VRCNet does not conduct experiments on PCN dataset. NSFA does not follow a coarse-to-fine framework, and GRNet uses a totally different baseline to get the refined shape, so we do not involves these two approaches. In addition, we try replacing the U-Net with PointNet++ autoencoder. Table 11 shows our transformer can also boost the performance with other frameworks, especially on Completion3D benchmark, and using the U-Net achieves a trade-off between performance and model size.

Methods	PCN dataset		Completion3D		Model Size
	Before	After	Before	After	
PointNet++ U-Net	9.67	9.15	12.15	10.67	31Mib
Wang et al.	8.51	8.27	10.70	8.97	61Mib
VRCNet	-	-	8.12	7.69	67Mib
Ours	8.63	8.24	8.65	7.78	54Mib

TABLE 11: Chamfer distance with our transformer applied to other frameworks.

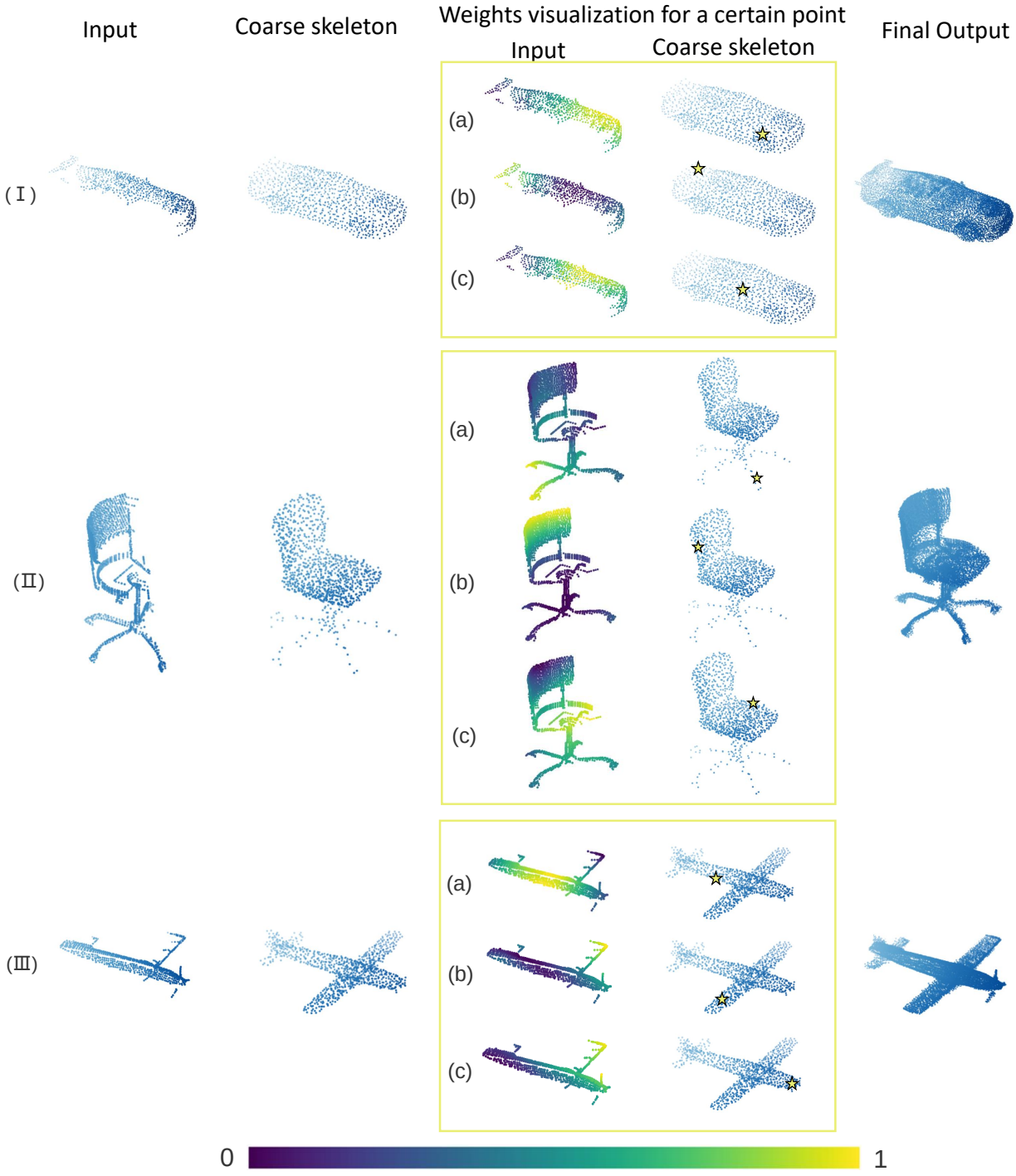


Fig. 10: Visualization of the weights for points in partial input to a query point in coarse skeleton when performing cross-attention.

10 Conclusion

In this paper, we have proposed a novel coarse-to-fine point cloud completion network leveraging the transformer model. The proposed skeleton-detail transformer effectively enhances the global shape with local geometric details by establishing the correlations between each other. We also consider the memory usage of the proposed transformer and thus propose a selective attention mechanism. The synthetic and real-world data experiments demonstrated our network's effectiveness in enhancing the geometric details in point cloud completion.

Acknowledgment

This work is partially supported by the Key Technological Innovation Projects of Hubei Province (2018AAA062), NSFC (No. 61972298), Wuhan University-Huawei GeoInformatices Innovation Lab.

References

[1] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3d-encoder-predictor cnns and shape synthesis," in *CVPR*, 2017, pp. 5868–5877.

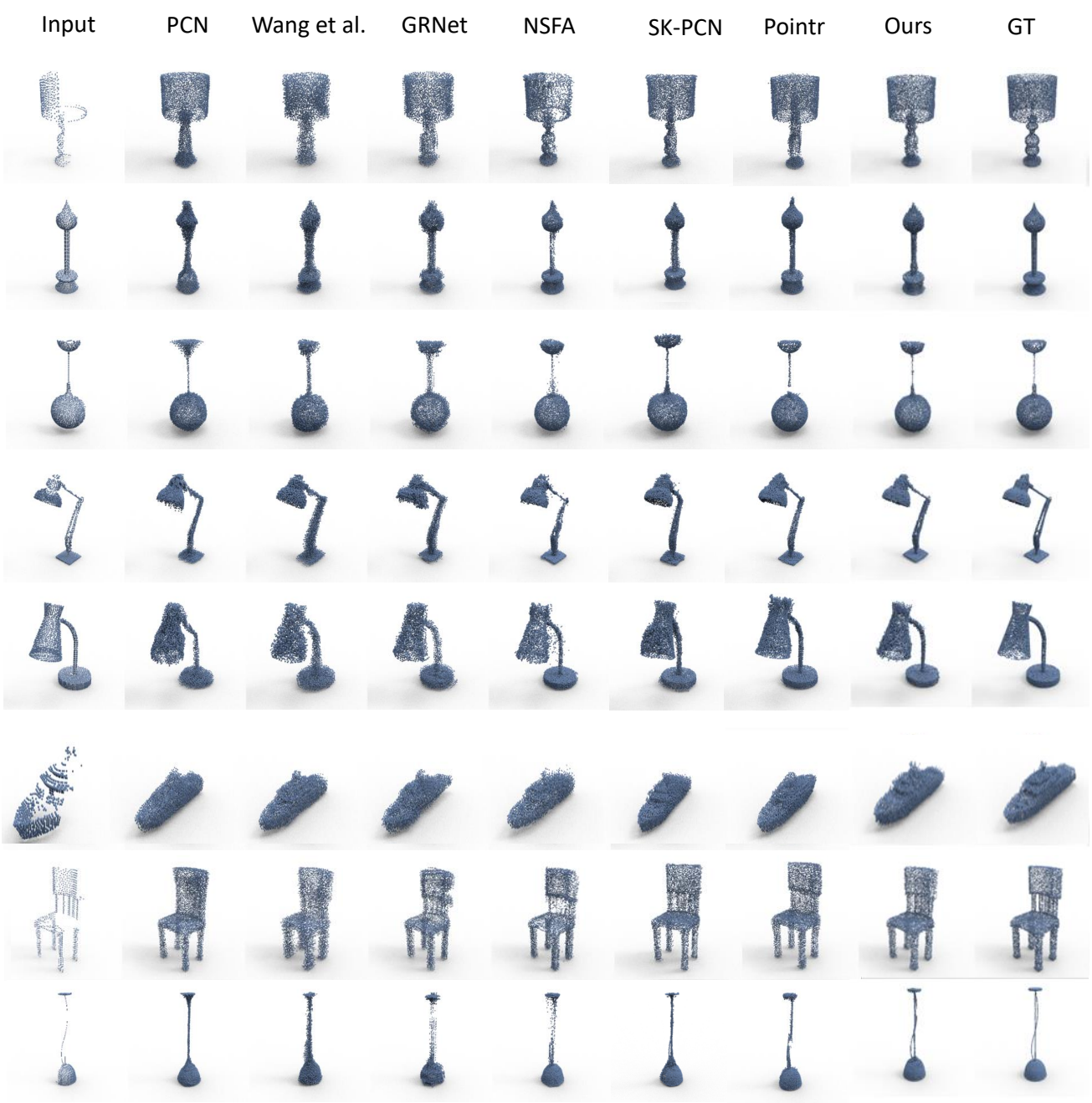


Fig. 11: Visual comparisons with state-of-the-art methods on ShapeNet dataset.

[2] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IROS*. IEEE, 2017, pp. 2442–2447.

[3] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans," in *CVPR*, 2018, pp. 4578–4587.

[4] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[5] Q. Yan, L. Yang, L. Zhang, and C. Xiao, "Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3836–3844.

[6] T. Cao, F. Luo, Y. Fu, W. Zhang, S. Zheng, and C. Xiao, "Dgecn: A depth-guided edge convolutional network for end-to-end 6d pose estimation," in *Conference on Computer Vision and Pattern Recognition*, 2022.

[7] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.

[8] D. Li, T. Shao, H. Wu, and K. Zhou, "Shape completion from a single rgbd image," *TVCG*, vol. 23, no. 7, pp. 1809–1822, 2016.

[9] X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu, "High-resolution shape completion using deep neural networks for global structure and local geometry inference," in *ICCV*, 2017, pp. 85–93.

[10] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3d object reconstruction from a single depth view with adversarial learning," in *ICCV*, 2017, pp. 679–688.

[11] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point

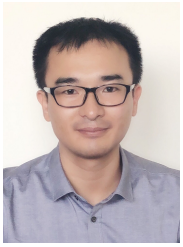
completion network," in *3DV*. IEEE, 2018, pp. 728–737.
[12] L. Pan, "Ecg: Edge-aware point cloud completion with graph convolution," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4392–4398, 2020.
[13] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 790–799.
[14] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "Grnet: gridding residual network for dense point cloud completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 365–381.
[15] 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*. Springer, 2020, pp. 512–528.
[16] 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*, 2021, pp. 7443–7452.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, p. 30.
[18] L. Pan, X. Chen, Z. Cai, J. Zhang, H. Zhao, S. Yi, and Z. Liu, "Variational relational point completion network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8524–8533.
[19] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 2021, p. online.
[20] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*. ACM, 2006, pp. 381–389.
[21] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proceedings Shape Modeling Applications,*. IEEE, 2004, pp. 191–199.
[22] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ToG*, vol. 32, no. 3, p. 29, 2013.
[23] S. Thrun and B. Wegbreit, "Shape from symmetry," in *ICCV*, vol. 2. IEEE, 2005, pp. 1824–1831.
[24] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," in *TOG*, vol. 27, no. 3. ACM, 2008, p. 43.
[25] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," in *TOG*, vol. 25, no. 3. ACM, 2006, pp. 560–568.
[26] Y. Li, A. Dai, L. Guibas, and M. Nießner, "Database-assisted object retrieval for real-time 3d reconstruction," in *CGF*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 435–446.
[27] Y. Shi, P. Long, K. Xu, H. Huang, and Y. Xiong, "Data-driven contextual modeling for 3d scene understanding," *Computers & Graphics*, vol. 55, pp. 55–67, 2016.
[28] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas, "Acquiring 3d indoor environments with variability and repetition," *TOG*, vol. 31, no. 6, p. 138, 2012.
[29] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "Topnet: Structural point cloud decoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.
[30] M. Sarmad, H. J. Lee, and Y. M. Kim, "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion," in *CVPR*, 2019, pp. 5898–5907.
[31] W. Zhang, C. Long, Q. Yan, A. L. Chow, and C. Xiao, "Multi-stage point completion network with critical set supervision," *Computer Aided Geometric Design*, vol. 82, p. 101925, 2020.
[32] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "Pf-net: Point fractal network for 3d point cloud completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7662–7670.
[33] 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*, vol. 34, no. 07, 2020, pp. 11 596–11 603.
[34] A. Richard, I. Cherabier, M. R. Oswald, M. Pollefeys, and K. Schindler, "Kaplan: A 3d point descriptor for shape completion," pp. 101–110, 2020.
[35] Z. Han, X. Wang, Y.-S. Liu, and M. 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)*. IEEE, 2019, pp. 10 441–10 450.
[36] X. Chen, B. Chen, and N. J. Mitra, "Unpaired point cloud completion on real scans using adversarial training," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020, p. online.
[37] R. Wu, X. Chen, Y. Zhuang, and B. Chen, "Multimodal shape completion via conditional generative adversarial networks," in *The European Conference on Computer Vision (ECCV)*, August 2020.
[38] 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*, 2021, pp. 12 498–12 507.
[39] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla, "Asfm-net: Asymmetrical siamese feature matching network for point completion," pp. 1938–1947, 2021.
[40] T. Huang, H. Zou, J. Cui, X. Yang, M. Wang, X. Zhao, J. Zhang, Y. Yuan, Y. Xu, and Y. Liu, "Rfnet: Recurrent forward network for dense point cloud completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 508–12 517.
[41] 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*, 2021, pp. 5499–5509.
[42] Y. Nie, Y. Lin, X. Han, S. Guo, J. Chang, S. Cui, J. Zhang *et al.*, "Skeleton-bridged point completion: From global inference to local adjustment," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 119–16 130, 2020.
[43] X. Wang, M. H. Ang, and G. H. Lee, "Voxel-based network for shape completion by leveraging edge generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 189–13 198.
[44] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
[45] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," pp. 2978–2988, 2019.
[46] F. Wu, A. Fan, A. Baevski, Y. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," in *International Conference on Learning Representations*, 2018, p. online.
[47] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, vol. 32, 2019.
[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020, p. online.
[49] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3464–3473.
[50] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Advances in Neural Information Processing Systems*, 2019, p. 32.
[51] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 076–10 085.
[52] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.
[53] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," pp. 16 259–16 268, 2021.
[54] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134 826–134 840, 2021.
[55] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," pp. 7463–7472, 2021.
[56] L. Hui, H. Yang, M. Cheng, J. Xie, and J. Yang, "Pyramid point cloud transformer for large-scale place recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6098–6107.
[57] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5301–5310.

[58] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
[59] L. Pan, C.-M. Chew, and G. H. Lee, "Pointatrousgraph: Deep hierarchical encoder-decoder with point atrous convolution for unorganized 3d points," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1113–1120.
[60] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *CVPR*, 2017, pp. 605–613.
[61] 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*, 2018, pp. 216–224.
[62] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, vol. 3, 2018.
[63] J. Zhang, W. Chen, Y. Wang, R. Vasudevan, and M. Johnson-Roberson, "Point set voting for partial point cloud analysis," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 596–603, 2021.
[64] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Softpoolnet: Shape descriptor for point cloud completion and classification," pp. 70–85, 2020.
[65] X. Wang, M. H. Ang, and G. Lee, "Cascaded refinement network for point cloud completion with self-supervision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
[66] W. Yifan, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3d point set upsampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5958–5967.
[67] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. 1, 2015.
[68] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
[69] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.
[70] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3405–3414.
[71] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

Wenxiao Zhang is now pursuing his Ph.D. degree in the School of Computer Science, Wuhan University, China. Before that he received his M.E. degree from Huazhong University of Science and Technology and B.E. degree from Shandong Normal University 2016 and 2014, respectively. His research interests include point cloud analysis, such as point cloud completion and point cloud based retrieval for place recognition.

Huajian Zhou received his Bachelor's Degree in the School of Computer Science, Wuhan University, China in 2014. Now he is working toward his master's degree in School of Computer Science, Wuhan University. His research interests are point cloud processing and texture recover.

Zhen Dong received his B.E. and Ph.D. degrees in RemoteSensing and Photogrammetry from the Wuhan University in 2011 and 2018. He is currently associate professor of theState Key Laboratory of Information Engineering in Surveying, Mapping andRemote Sensing (LIESMARS), Wuhan University. His research interests include 3D vision, point clouds processing and deep learning.



Jun Liu received the PhD degree from Nanyang Technological University, the MSC degree from Fudan University, and the BEng degree from Central South University. His research interests include computer vision and artificial intelligence. His works have been published in premier computer vision journals and conferences, including TPAMI, CVPR, ICCV, and ECCV. He received the Best Thesis Award from EEE, Nanyang Technological University, and is listed in the top 2% scientists worldwide identified by Stanford University in 2021. He is an Associate Editor of IEEE Transactions on Image Processing, and Area Chair of ICML 2022, NeurIPS 2022, ICLR 2022, and WACV 2022.

Qingan Yan received his Ph.D. degree at Computer Science Department of Wuhan University in 2017. Before that, he got his M.S. and B.E. degree in Computer Science from Southwest University of Science and Technology, and Hubei Minzu University, in 2012 and 2008 respectively. He is working at OPPO US Research Center. Before that he served as a research scientist at JD.com Silicon Valley Research Center in California from 2017 to 2021. His research interests include 3D reconstruction, geometric scene perception, visual correspondence and image synthesis.

Chunxia Xiao received the B.Sc. and M.Sc. degrees from the Department of Mathematics, Hunan Normal University, in 1999 and 2002, respectively, and the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University, in 2006, China. He became an Assistant Professor at Wuhan University in 2006, and became a Professor in 2011. During October 2006 to April 2007, he worked as a postdoc at Hong Kong University of Science and Technology, and During February 2012 to February 2013, he visited University of California-Davis for one year. His research areas include computer graphics, computer vision, virtual reality, and augmented reality. He has published more than 110 papers in journals and conferences. He is a member of IEEE and ACM.