PCTMA-Net: Point Cloud Transformer with Morphing Atlas-based Point Generation Network for Dense Point Cloud Completion

Jianjie Lin, Markus Rickert, Alexander Perzylo and Alois Knoll

Abstract— **Inferring a complete 3D geometry given an incomplete point cloud is essential in many vision and robotics applications. Previous work mainly relies on a global feature extracted by a Multi-layer Perceptron (MLP) for predicting the shape geometry. This suffers from a loss of structural details, as its point generator fails to capture the detailed topology and structure of point clouds using only the global features. The irregular nature of point clouds makes this task more challenging. This paper presents a novel method for shape completion to address this problem. The Transformer structure is currently a standard approach for natural language processing tasks and its inherent nature of permutation invariance makes it well suited for learning point clouds. Furthermore, the Transformer's attention mechanism can effectively capture the local context within a point cloud and efficiently exploit its incomplete local structure details. A morphing-atlas-based point generation network further fully utilizes the extracted point Transformer feature to predict the missing region using charts defined on the shape. Shape completion is achieved via the concatenation of all predicting charts on the surface. Extensive experiments on the Completion3D and KITTI data sets demonstrate that the proposed PCTMA-Net outperforms the state-of-the-art shape completion approaches and has a 10% relative improvement over the next best-performing method.**

I. Introduction

The use of point clouds as a format of shape representation has increased in the last years due to the rapid development of 3D acquisition technologies such as Lidar and depth cameras. The limited sensor resolution, occlusion, and camera angles however make it challenging to obtain a point cloud representation of the complete shape of an object. As a result, the acquired raw points are typically sparse, noisy, and miss large regions. On the other hand, complete 3D shapes are essential in vision applications, such as semantic segmentation and SLAM [1]. A complete 3D shape can improve the performance of CAD model-based point registration [2] and enables more flexible grasp planning [3], [4]. In this work, we focus on completing partial 3D shapes that suffer from occlusion and limited sensor resolution.

Previous work [5], [6], [7] principally followed the encoder-decoder paradigm framework by extracting a latent global feature from an incomplete point cloud. Decoders leverage these feature to predict missing regions. Benefiting from PointNet-based [8] feature extractor networks, the task of shape completion made tremendous progress in recent years. However, the extracted global features from PointNet ignore the geometric relationship within the point clouds due to the max-pooling operation. As a result, these approaches suffer from a loss of structural detail in the reconstruction.

The intuitive solution is to make up for the shortcomings of the PointNet by excavating the semantic affinity within the point cloud. Therefore, we propose a novel framework named Point Cloud Transformer with Morphing-Atlas-based Point Generation Network for Shape Completion (PCTMA-Net) to address this problem. The Transformer [9] is a standard framework for natural language processing and has been further extended to vision tasks for image recognition [10], as well as point cloud classification and segmentation [11]. The Transformer follows the encoder-decoder structure and consists of four main modules: input embedding, positional encoding, (self-)attention mechanism, and positional feed-forward. In this work, we apply only the encoder module and neglect the positional encoding module due to the point cloud's irregular nature. The Transformer's central core is the attention mechanism, which can generate refined attention features by leveraging the global context. The attention weight between any two positions is updated by the dot product of query and key vector. The weighted sum of all attention weights is the attention feature. The concept of query, key, and value vector makes it possible to match and learn the global context. The attention feature of each word is related to all input features. Furthermore, the permutation invariant nature of softmax, dot product, and point-wise feed-forward network makes it well-suited for point cloud learning. The offset attention mechanism introduced in [11] uses the idea of the Laplacian matrix to improve the attention performance further. In this work, we replace the original attention design with the offset attention mechanism. The morphing-atlas-based point generation network is the decoder component in our overall structure. The extracted global feature from the Transformer is further utilized to generate the points. An atlas, as defined in topology, consists of a set of charts on a surface. Therefore, we assume that a missing region of the surface can be recovered by a chart. Based on this assumption, we duplicate the Transformer feature and concatenate it with a predefined grid. We utilize the idea of multi-head attention by linearly projecting the concatenated features to learn n_{chart} different features, where each feature is responsible for generating a chart defined on the surface. We quantitatively and qualitatively evaluated the proposed PCTMA-Net on the Completion3D data set and demonstrate a 10% relative improvement over the next best-performing method for the task of shape completion.

Jianjie Lin, Markus Rickert, Alexander Perzylo, Alois Knoll are with Robotics, Artificial Intelligence and Real-time Systems, Department of Informatics, Technische Universität München, Munich, Germany {jianjie.lin, rickert, perzylo,knoll}@in.tum.de

Furthermore, the qualitative evaluation on the KITTI data set shows that our proposed network is able to predict more structural details than other state-of-the-art approaches.

Our contributions are summarized as follows: (1) We propose a novel shape completion framework named Point Cloud Transformer with Morphing-Atlas-based point generation Network for shape completion (PCTMA-Net), which is inherently permutation-invariant and has the capability of learning the global context within the point clouds and preserving structural details. (2) The integration of the concept of an atlas and the multi-head attention mechanism leads to the generation of high-resolution, high-fidelity, and fine-grained shapes. (3) Extensive experiments are conducted on the Completion3D benchmarks, and the KITTI data set, which indicate that the proposed networks remarkably outperforms other competitive methods.

II. Related Work

Shape completion approaches made significant progress in recent years due to the rapid development of deep learning and 3D acquisition technologies. We can roughly categorize the existing work into volumetric-based and multilayer perceptron-based networks from the perspective of network structure and the underlying 3D data representation.

Volumetric-based shape completion: The extension of CNN to 3D convolutional neural networks can be used for dealing with a shape in the volumetric representation [12], [13]. Notable work such as 3D-Encoder-Predictor Networks (3D-EPN) [14] progressively reconstruct the 3D volumetric shape. The work in [15] directly generates the high resolution 3D volumetric shape by combining the global structure with the refinement of local geometry, while [16] introduced a variational auto-encoder to learn a shape prior to inferring the latent representation of complete shapes. GRNet [17] took one step further by introducing Gridding and Gridding Reverse to convert between point clouds and 3D grids. However, a quantization effect is introduced during the transformation of point clouds into a 3D volumetric representation. The computational costs increase cubically to the resolution and therefore make it more challenging to process fine-grained shapes.

Multilayer perceptron (MLP)-based shape completion: Point clouds can be directly obtained by several acquisition techniques. It is much more efficient compared to the voxel-based representation when processing costs are compared. Inspired by PointNet [8] and its successor work [18],[19], several approaches use them for point cloud learning, as the point-wise MLP enables the handling of irregular point clouds and aggregating features using a symmetric function. However, the PointNet network suffers from a loss of structure details. The current state-of-the-art approaches for shape completion such as AtlasNet [6], PCN [20] and Folding-Net [7] use PointNet as their baseline to extract global features and to apply a decoder to predict the missing regions. Unlike PCN and FoldingNet, AtlasNet completes the shape by generating surface elements utilizing the atlas concept. TopNet [5] improves the decoder by using a hierarchical rooted tree. By combining reinforcement learning with an adversarial network, RL-GAN-Net [21] and Render4Completion [13] propose a reinforcement learning agent-controlled GAN to improve the quality and consistency of the generated complete shape. However, most of these studies suffer from information loss on structural details, as they predict the whole point cloud only from a single global shape representation. SA-Net [22] extended these approaches with a skip-attention mechanism to preserve more structural details. PF-Net [23] introduced a point pyramid decoder to generate a shape in different resolution levels.

III. The Architecture of PCTMA-Net

A. Overview

The overall structure of PCTMA-Net is illustrated in Fig 1, which aims to learn a semantic affinity within a partial point cloud by using a Transformer encoder. The complete 3D shape is reconstructed with a morphing-atlas decoder utilizing the extracted feature from the Transformer encoder. We formulate the whole shape completion pipeline as: Given a partial point cloud, indicated as $\mathcal{P}$ with N_{in} points, where each point is represented in 3D coordinates $\mathbf{x} = [x_i, y_i, z_i]$, we first convert this partial point cloud into a feature vector $\mathbf{F}_0$ by a PointNet. The difference to previous work [7], [6], which relies on only the global feature for shape completion, is that we further utilize the Transformer encoder to process the feature to obtain a piece of semantic affinity information for predicting the missing regions. The extracted feature is later fed to the morphing-atlas point generator for completing the shape.

B. Point Cloud Transformer Encoder

The Transformer encoder of PCTMA-Net first transforms an incomplete point cloud to the feature space using an input embedding network. We then feed the extracted feature to $N\times$ stacked encoder layers, where they share a similar philosophy of design as the original paper [9], except for the attention mechanism. The purpose of the encoder layer is to learn a discriminate representation for each point. The encoder can be mathematically formulated in the following: By given a partial point cloud $\mathcal{P} \in \mathcal{R}^{N_{\text{in}}\times d}$ with N_{in} points each having a d-dimensional feature description, an embedding feature $\mathbf{F}_0$ is firstly learned with an input embedding network, indicated as $\text{F}_{\text{embedding}}$. The difference to the embedding network presented in [11] is that we defined $\text{F}_{\text{embedding}}$ as a PointNet followed by a max-pooling operator. As a result, we acquire a d_{model}-dimensional embedding feature $\mathbf{F}_0 \in \mathcal{R}^{d_{\text{model}}}$ instead of $\mathbf{F}_0 \in \mathcal{R}^{d_{\text{model}}\times N_{\text{in}}}$[11]. It will improve the shape completion performance, as the $\mathbf{F}_0$ after max-pool operator can reduce redundant information and make the training more efficient. The global feature $\mathbf{F}_0$ is later fed to $\text{F}_{\text{encoder}_i}$:

$$\mathbf{F}_i = \text{F}_{\text{encoder}_i}(\mathbf{F}_{i-1}),\ i = [1, \dots, N]\,. \tag{1}$$

Furthermore, we concatenate the features from each encoder layer and follow up by two cascade LBR layers to form an

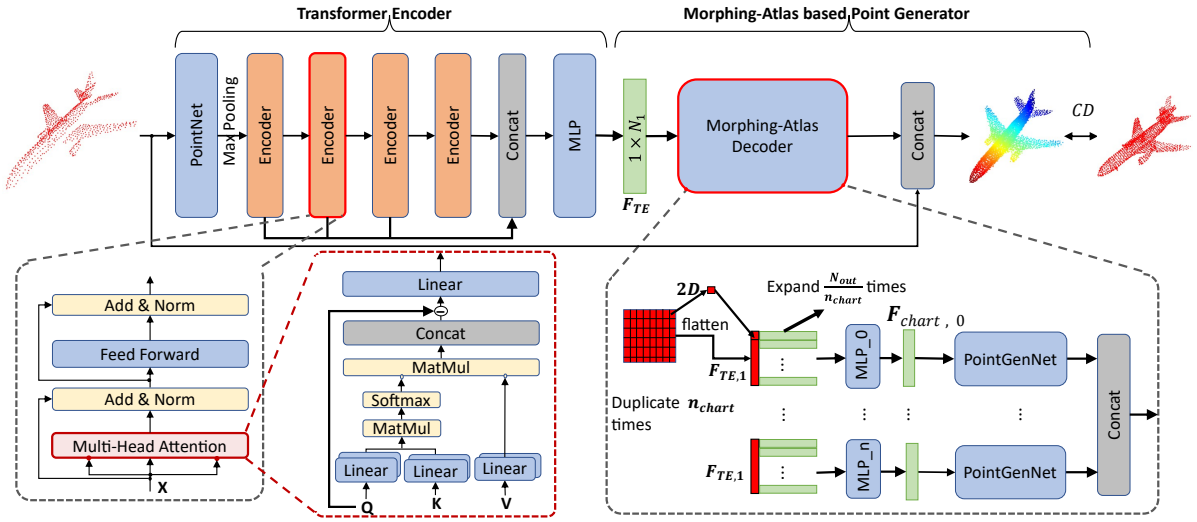

Fig. 1: The overall structure of PCTMA-Net. The whole structure consists of a Transformer encoder and morphing-atlas point generation decoder. The Transformer encoder aims to extract features from the input point clouds by using an $N\times$ stacked encoder layer which consists of an attention mechanism and positional feed-forward network. The morphing-atlas-based surface reconstruction decoder uses multi-chart point generation networks for point cloud completion by concatenating the features from the Transformer encoder and mesh grid.

effective global feature

$$\mathbf{F}_{\mathrm{e}} = \mathrm{BatchNorm}(\mathbf{F}_i \oplus \cdots \oplus \mathbf{F}_N) \tag{2}$$

$$\mathbf{F}_{\mathrm{TE}} = \mathrm{LBR}\big(\mathrm{LBR}(\mathbf{F}_{\mathrm{e}})\big)\,, \tag{3}$$

where $\mathbf{F}_{\mathrm{i}} \in \mathcal{R}^{d_{\mathrm{model}}}$, $\mathbf{F}_e \in \mathcal{R}^{N\times d_{\mathrm{model}}}$ and $\mathbf{F}_{\mathrm{TE}} \in \mathcal{R}^{d_{\mathrm{model}}}$. The operator $\oplus$ is denoted as concatenation, and the function LBR represents a linear layer followed by BatchNorm and ReLU operators. The F_{encoder_i} consists of two sub-layers, namely self-attention mechanism and positional forward feedback:

$$\mathrm{F}_{\mathrm{encoder}_i}\big(\mathbf{F}_{i-1}\big) = \mathrm{FFN}_i\big(\mathrm{attention}_i(\mathbf{F}_{i-1})\big), \tag{4}$$

$$\mathrm{FFN}_i(\mathbf{x}) = \mathrm{LBR}_{i,1}\big(\mathrm{LBR}_{i,0}(\mathbf{x})\big) + \mathbf{x}\,. \tag{5}$$

The layer FFN_i is a shared positional forward feedback network comprising two cascaded LBRs with the size of $[d_{\mathrm{ff}}, d_{\mathrm{model}}]$, where $d_{\mathrm{ff}} = 2048$ and $d_{\mathrm{model}} = 1024$.

a) Offset self-attention mechanism: Self-attention is a mechanism that calculates the semantic relationship between different elements within a sequence of data. In the context of point cloud processing, attention is employed to build weights between every two positions in the feature space. In comparison to k-nearest neighbors algorithms, the attention mechanism has a larger receptive field. Furthermore, the attention mechanism's permutation invariant property makes it suitable for disordered, irregular data representation such as point clouds. The work in [11] proposed the offset attention by utilizing the idea of a Laplacian matrix $L = D - E$, where E is the adjacent matrix E and D is the diagonal matrix. The attention mechanism is adopted as

$$\begin{aligned}\mathbf{F}_{\mathrm{sa,out}} &= \mathrm{attention}(\mathbf{F}_{\mathrm{sa,in}}) \\ &= \mathrm{LBR}(\mathbf{F}_{\mathrm{sa,in}} - \mathbf{F}_{\mathrm{sa}}) + \mathbf{F}_{\mathrm{in}}\,.\end{aligned} \tag{6}$$

The remaining part of the attention computation operators still follows the same design as in the original paper [9]. The self-attention feature $\mathbf{F}_{\mathrm{sa}}$ in (6) concatenates the multi-head attention with the following formulation:

$$\mathbf{F}_{\mathrm{sa}} = \mathrm{Linear}(\mathbf{F}_{\mathrm{head}_1} \oplus \cdots \oplus \mathbf{F}_{\mathrm{head}_h})\,, \tag{7}$$

where the attention feature at the i-head $\mathbf{F}_{\mathrm{head}_i}, i \in [1, \ldots, h]$ is formulated as

$$\mathbf{F}_{\mathrm{head}_i} = \mathrm{softmax}\Big(\frac{\hat{\mathbf{Q}}\hat{\mathbf{K}}^T}{\sqrt{d_k}}\Big)\hat{\mathbf{V}}\,, \tag{8}$$

with $\hat{\mathbf{Q}} = \mathrm{Linear}(\mathbf{Q})$, $\hat{\mathbf{K}} = \mathrm{Linear}(\mathbf{K})$, $\hat{\mathbf{V}} = \mathrm{Linear}(\mathbf{V})$. The variables $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ are projected with a different linear layer, respectively. Following the same principle as the original paper, we set $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{F}_{\mathrm{sa,in}} \in \mathcal{R}^{d_{\mathrm{model}}}$. We reshape the linear projected query and key as $\hat{\mathbf{Q}}, \hat{\mathbf{K}} \in \mathcal{R}^{d_{\mathrm{model}}\times 1}$ to obtain the attention weights $\mathbf{A}$ by matrix dot product via $\hat{\mathbf{Q}}\hat{\mathbf{K}}^T$. We normalize $\mathbf{A}$ with $\sqrt{d_k}$ to avoid large values in magnitude, where $d_k = \frac{d_{\mathrm{model}}}{h}$. The equation in (8) shows, that the self-attention $\mathbf{F}_{\mathrm{head}_i}$ is equal to the weighted sums of the value vector $\mathrm{Linear}(\mathbf{V})$ using the corresponding attention weights. The multi-head attention mechanism can jointly capture information from different representation subspace at different positions [9]. Therefore, it can efficiently preserve and capture the point cloud's detailed topology and structure for predicting the missing regions in comparison to [5], [6].

C. Morphing-Atlas-Based Point Generation Network

At the first stage, the Transformer encoder extracts a global feature $\mathbf{F}_{\mathrm{TE}}$ for expressing an incomplete point cloud. We then feed the extracted features into a morphing-atlas-based point generator for predicting continuous and smooth

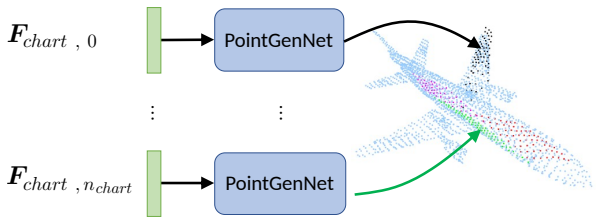

Fig. 2: Visualization of MA-Network

shapes. Atlas [6] is defined in the topology for describing a manifold and an atlas is composed of each chart that, roughly speaking, describes the local region of the manifold. In the context of 3D shapes, the manifold can be considered as a shaped surface. Therefore, we can represent a 3D shape by combing all the charts. Based on the Atlas concept, we define a chart as $\mathcal{C}_i$ and let a designed decoder $\mathcal{D}_i$ learn to map a 2D grid to a 3D surface. Furthermore, we introduce a hyper-parameter n_{chart} to control the number of charts defined on a shape to predict a smooth and high-resolution shape. The global feature $\mathbf{F}_{\text{TE}} \in \mathcal{R}^{d_{\text{model}}}$ is duplicated $N_{\text{out}}/n_{\text{chart}}$ times and then concatenated with a mesh grid to describe a new feature, denoted as $\mathbf{F}_{\text{TE},1} \in \mathcal{R}^{(d_{\text{model}}+2)\times(N_{\text{out}}/n_{\text{chart}})}$. It beneficial to linearly project $\mathbf{F}_{\text{TE},1}$ with different learned linear projections. This concept is similar to multi-head attention by allowing the model to obtain the shape features from different representation subspaces at different positions. Therefore, $\mathbf{F}_{\text{TE},1}$ is duplicated n_{chart} times and each $\mathbf{F}_{\text{TE},1}$ is fed to an MLP layer which produces a new hidden code, denoted as $\mathbf{F}_{\text{chart},i} \in \mathcal{R}^{(d_{\text{model}}+2)\times(N_{\text{out}}/n_{\text{chart}})}, i \in [1,\dots,n_{\text{chart}}]$. For each single chart, we feed $\mathbf{F}_{\text{chart},i}$ into a PointGenNetwork (Fig. 2), sharing the same structure as in [6]. All charts are concatenated to form a complete shape.

D. Evaluation Metrics

We apply the Chamfer distance (CD) [24] as a quantitative evaluation metric due to its efficient computation compared to the earth mover's distance [24]. The Chamfer distance measures the mean distance between each point in one point cloud to its nearest neighbor in another point cloud. Let $S_G = [x_i, y_i, z_i]_i^{n_G}$ be the ground truth and $S_R = [x_i, y_i, z_i]_i^{n_R}$ be the reconstructed point by given a partial point cloud. n_G and n_R indicate the number of points in S_G and S_R, respectively. The Chamfer distance d_{CD} of S_G and S_R with $L2$ norm is formulated as

$$d_{\text{CD}} = \frac{1}{n_R}\sum_{x\in S_R}\min_{y\in S_G}||x-y||^2 + \frac{1}{n_G}\sum_{y\in S_G}\min_{x\in S_R}||x-y||^2. \quad (9)$$

E. Implementation details

We implemented PCTMA-Net in PyTorch, where the model is optimized with an Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, together with a CosineAnnealingLR scheduler. The number of encoder layers used in the Transformer encoder is set to 4, and we follow the original papers by setting the multi heads in the offset attention mechanism to 8. We trained the network on a Linux system with a 2.6 GHz Intel Core i7–6700HQ, 16 GB of RAM, and one Nvidia RTX 2080 Ti GPU.

IV. Experiments

We compare our proposed shape completion algorithm PCTMA-Net with other state-of-the-art approaches on two large scale data sets: Completion3D [5] and KITTI [26]. The Chamfer distance is employed as a metric in the evaluation.

A. Shape Completion on Completion3D Data Set

Completion3D [5] from ShapeNet [27] offers a data set, which consists of 28 974 training samples and 800 point cloud evaluation samples with a point resolution of 2048 for training and validation, respectively. In the comparison, we use different output resolutions and the quantitative results are summarized in Table I. Note that the results of FoldNet [7], SA-Net [22], and PCN [20] are cited from the Completion3D benchmark leaderboard. Table I shows that our PCTMA-Net algorithm outperforms the other methods in 6 out of 8 categories with the overall Chamfer distance of 9.48 for $N_{\text{out}} = 16\,152$ and $n_{\text{chart}} = 32$. The qualitative visualization of completion results shown in Fig. 3 indicates that our approach is able to predict more details. The performance in the quantitative and qualitative evaluations proves the Transformer encoder and the morphing-atlas decoder's effectiveness for predicting and preserving the shape details.

B. Shape Completion on Robustness of Input Resolution

The input resolution can greatly affect the performance of a neural network. In this section, we will study the robustness of input resolution on the different network structures. We downsample the evaluation data set from Completion3D to obtain four levels of input resolutions: 256, 512, 1024, and 2048. The visualization of these four levels of input resolutions is shown in Fig. 4a. All networks are trained on an input resolution of 2048 and output a fixed size of 16 384 points. For point resolutions less than 2048, we follow the principle in PCN [20] to select points from the input randomly and pad the input cloud to raise the number of points to 2048. We evaluate these four levels of input resolution on the Completion3D data set. The quantitative illustration in Fig. 4b indicates that our network has the best robustness and outperforms the other approaches in all four input resolutions experiments.

C. Shape Completion on KITTI data set

For a further study of the application area, we conduct experiments on the KITTI data set [26], which is collected from real-world Velodyne Lidar scans composed of 2401 highly sparse point clouds. Note that the KITTI data set does not include the ground truth in a quantitative evaluation. Therefore, we can only qualitatively visualize the shape completion results. Unlike other work [5], [17], which trains the network with only the car category in ShapeNet [27] and then evaluates the KITTI data set, we use the same trained network as in Section IV-A for evaluation. This evaluation strategy can show the capability of the generalization of

TABLE I: Point completion results on Completion3D with ground truth and input resolution (2048 points) compared using Chamfer distance (CD) with L^2 norm. The results are multiplied by 10^4. In our algorithm (PCTMA-Net), we set meshgrid = 0.05. The best result is highlighted in ***green***, and a lower value is better.

Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Overall
AtlasNet ($N_{\text{out}} = 2048$) [6]	5.82	29.28	11.02	27.11	34.04	19.11	29.27	15.55	21.40
AtlasNet ($N_{\text{out}} = 16384$) [6]	5.50	19.89	9.23	21.17	30.99	15.34	21.67	14.64	17.31
FoldNet [7]	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51	19.07
FCN [20]	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73	18.22
TopNet ($N_{\text{out}} = 16384$) [5]	5.85	21.27	10.03	20.09	22.98	14.65	24.25	11.78	16.36
PointNetFCAE ($N_{\text{out}} = 2048$) [25]	5.81	21.14	8.95	22.01	33.36	15.81	27.52	14.09	18.59
PointNetFCAE ($N_{\text{out}} = 16384$) [25]	4.00	16.70	6.24	14.63	18.15	10.99	15.77	8.55	11.88
SA-Net [22]	5.27	14.45	7.78	13.67	13.53	14.22	11.75	8.84	11.22
GRNet ($N_{\text{out}} = 2048$) [17]	7.64	24.06	12.02	24.62	28.73	18.85	32.90	12.48	20.16
GRNet ($N_{\text{out}} = 16384$) [17]	3.79	14.86	6.71	12.74	13.73	11.05	15.43	**6.50**	10.60
Ours ($n_{\text{chart}} = 32$, $N_{\text{out}} = 2048$)	3.60	14.67	7.03	14.04	20.61	10.66	18.01	7.62	12.03
Ours ($n_{\text{chart}} = 128$, $N_{\text{out}} = 10240$)	**3.16**	13.53	6.58	13.21	12.93	10.29	14.25	6.98	10.11
Ours ($n_{\text{chart}} = 32$, $N_{\text{out}} = 16152$)	3.38	**13.00**	**6.12**	**12.72**	**11.87**	**9.18**	**12.43**	7.17	**9.48**

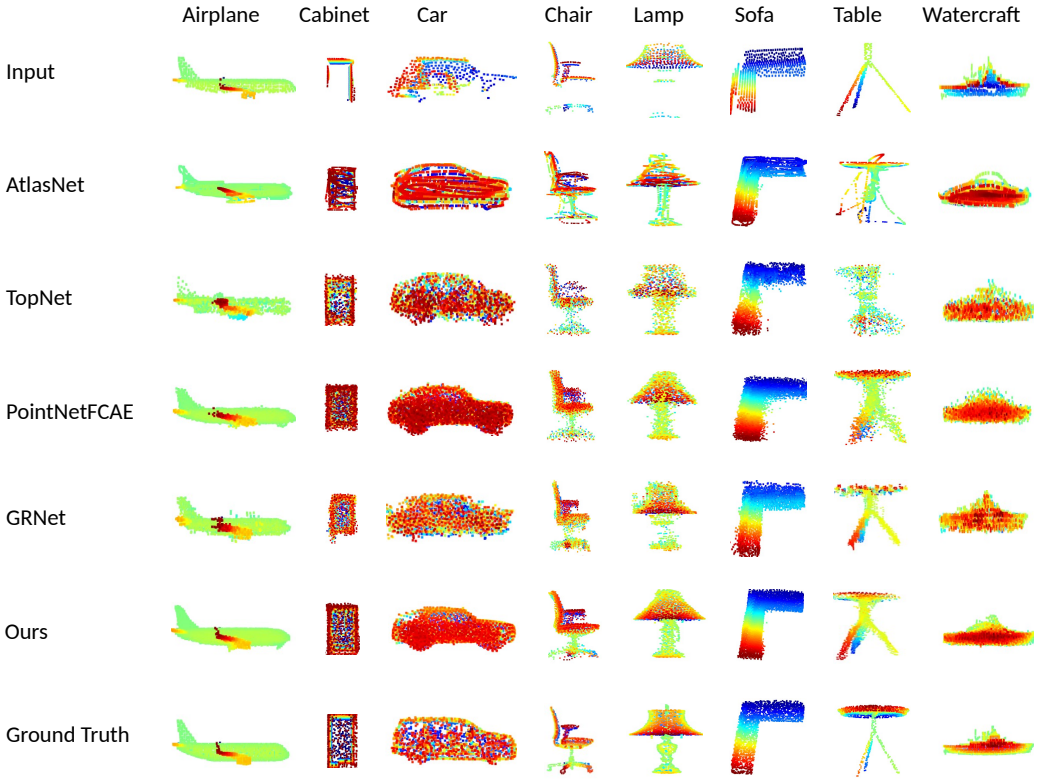

Fig. 3: Visualization of completion results on the Completion3D evaluation set.

one network. The incomplete point clouds from KITTI have diverse input resolutions and are highly sparse. We use the same strategy as in Section IV-B to lift the number of points to 2048. Besides, we transform the incomplete point cloud by using the 3D bounding boxes to get a point cloud that is distributed between $[-0.5, 0.5]$. The qualitative result illustrated in Fig. 5 indicates that our approach and PointNetFCAE can generate more detailed shape information compared to the other methods.

D. Ablation Studies

In this section, we will study the effectiveness of our designed structure and chosen hyper parameters. All studies are conducted on the Completion3D data set for consistency. Without loss of generality and without special instructions, we set $N_{\text{out}} = 10\,240$ and $n_{\text{chart}} = 32$ in the following

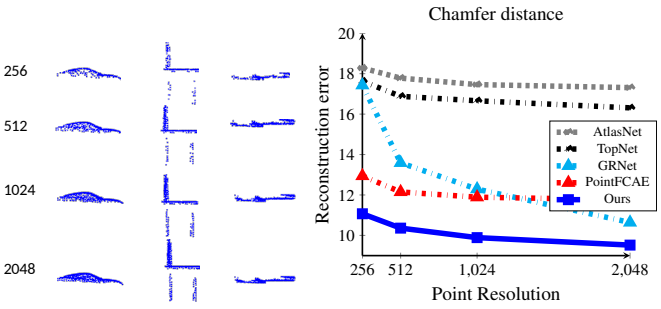

Fig. 4: In (a) the point resolution varies from 256, 512, 1024 to 2048. In (b), we compare the proposed approach against other state-of-the-art approaches on the Completion3D benchmarks. Lower values are better.

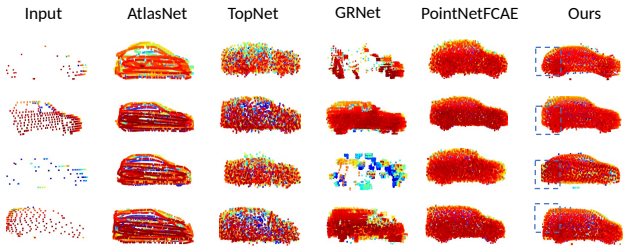

Fig. 5: Qualitative completion results on the LiDAR scans from KITTI. The incomplete input point cloud is extracted and normalized from the scene with its 3D bounding box.

experiment.

*1) **Effect of Transformer encoder**:* The Transformer encoder is the main core used in PCTMA-Net, which has two hyper parameters: the number of encoder layers n_{encoder} and the number of heads h used in the attention mechanism. In this section, we will study the effect on shape completion by varying different combinations of these two parameters. We can conclude from Table II, that we can achieve better shape completion performance with higher numbers of h and n_{encoder}. Taking various factors such as the network parameters into consideration, we set these hyper parameters to $h = 8$ and $n_{\text{encoder}} = 4$.

*2) **Effect of number of charts**:* The hyper parameter n_{char} is used to control the number of charts defined on a shape. In this section, we will study the effectiveness of the number of charts. We summarize the results in Table III. It can be shown, that PCTMA-Net can result in a smaller Chamfer distance with a greater number of charts. However, the parameters of the network will be increased correspondingly, which is shown in the second row of Table III.

TABLE II: The Chamfer distance (CD) on different hyper parameters in the Transformer encoder.

n_{encoder}	2		4		6	
h	4	8	4	8	4	8
CD ($\times 10^4$)	10.86	10.41	10.59	10.21	10.69	10.21

TABLE III: The Chamfer distance (CD) on the number of charts.

n_{chart}	8	32	128
CD ($\times 10^4$)	10.45	10.21	10.11
parameter ($\times 10^6$)	52.75	93.26	258.73

TABLE IV: The Chamfer distance (CD) on different grid types. In Meshgrid (k), k indicates the grid scale.

Grid type	Rand grid	Meshgird (0.5)	Meshgrid (0.05)
CD ($\times 10^4$)	11.36	10.25	10.21

*3) **Effect of grid strategy**:* In our proposed morphing-atlas decoder, the pointGenNet maps 2D grids to 3D surfaces. In this section, we will use the plane grid for point generation, which introduces two additional values. We can either randomly sample the value from $[0, 1]$ or use a grid with a predefined grid scale and grid size. The evaluation results on different grid strategies are listed in Table IV. It can be shown, that the mesh grid method shows significantly better performance in comparison to the randomly sampled grid methods. We further study the effectiveness of the grid scale by using the same grid size. The results in Table IV show that the mesh grid scale from 0.05 to 0.5 shares a similar performance.

*4) **Effect of metrics**:* Most existing work employs the Chamfer distance as a loss function due to its efficient computation. The earth mover's distance (EMD) is another option for point clouds and can be formulated as:

$$d_{\text{EMD}}(S_R, S_G) = \frac{1}{|S_G|} \min_{\Phi: S_R \to S_G} \sum_{X \in S_R} ||x - \Phi(x)||_2 , \quad (10)$$

where Φ is the bijection function. In this section, we will study the effect on shape completion of different training loss functions. The comparison results in Table V demonstrate, that for a pure EMD loss function, the shape completion value with the metric of CD has the worst performance. The utilization of CD and EMD in the loss function can reduce the Chamfer distance value, and generate a more uniformly distributed point cloud than the pure CD loss function. As EMD uses the bijection function to force the output to have the same density distribution as the ground truth for coping with the linear assignment problem. It hence can generate a point cloud which is more discriminative to local details. However, EMD is much more computationally expensive

TABLE V: The Chamfer distance (CD) on different loss functions.

Loss function	EMD	CD+EMD	CD
CD ($\times 10^4$)	16.12	10.45	10.21

TABLE VI: The Chamfer distance (CD) on different point generators. We abbreviate our Encoder as TE and connect to different algorithm point generators.

Methods	TE-FoldNet	TE-TopNet	TE-AtlasNet
CD ($\times 10^4$)	13.22	13.49	11.36

with approximately $\mathcal{O}(n^2)$, where n is the number of point cloud, compared to CD.

5) ***Effect of point generator****:* In this section, we study the effect of different point generators on shape completion, introduced in FoldNet [7], TopNet [5], by attaching them to our Transformer encoder. The results are summarized in Table VI. All of these three networks have improved to some degree by using the Transformer encoder. FoldNet shows an improvement from 19.07 to 13.22, TopNet improved from 16.36 to 13.49, and the performance of AtlasNet improved from 17.31 to 11.36.

V. Conclusion

We propose a novel network named PCTMA-Net for point cloud completion. Through its encoder-decoder structure, PCTMA-Net can effectively capture features of local regions for predicting missing shape parts. The utilization of the concept of an atlas further helps the network to reconstruct a smooth shape with a predefined number of charts. We conducted extensive experiments on the Completion3D and KITTI data sets to validate our proposed network structure's effectiveness. Via the experiments, we can conclude that our approach outperforms other state-of-the-art approaches on these two large data sets.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. Lin, M. Rickert, and A. Knoll, "6D pose estimation for flexible production with small lot sizes based on CAD models using gaussian process implicit surfaces," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[3] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2442–2447.

[4] J. Lin, M. Rickert, and A. Knoll, "Grasp planning for flexible production with small lot sizes based on CAD models using GPIS and bayesian optimization," 2021.

[5] L. P. Tchapmi, 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 (CVPR)*, June 2019.

[6] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry, "AtlasNet: A papier-mâché approach to learning 3D surface generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] 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*, 2018, pp. 206–215.

[8] 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*, 2017, pp. 652–660.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[10] 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," *arXiv preprint arXiv:2010.11929*, 2020.

[11] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *arXiv preprint arXiv:2012.09688*, 2020.

[12] 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 *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 679–688.

[13] T. Hu, Z. Han, A. Shrivastava, and M. Zwicker, "Render4Completion: Synthesizing multi-view depth maps for 3D shape completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 4114–4122.

[14] A. Dai, C. Ruizhongtai Qi, and M. Nießner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5868–5877.

[15] 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 *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 85–93.

[16] D. Stutz and A. Geiger, "Learning 3D shape completion under weak supervision," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1162–1181, 2020.

[17] 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.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2017, pp. 5105–5114.

[19] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions On Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[20] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2018.

[21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5898–5907.

[22] 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*, 2020, pp. 1939–1948.

[23] 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.

[24] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[25] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2018, pp. 40–49.

[26] 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.

[27] A. X. Chang, T. A. Funkhouser, L. J. 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," *CoRR*, vol. abs/1512.03012, 2015.