International Journal of Computer Vision (2023) 131:2425–2445
https://doi.org/10.1007/s11263-023-01820-y

MANUSCRIPT

Learning Geometric Transformation for Point Cloud Completion

Shengping Zhang[1] · Xianzhu Liu[1] · Haozhe Xie[2] · Liqiang Nie[3] · Huiyu Zhou[4] · Dacheng Tao[5] · Xuelong Li[6]

Received: 7 July 2022 / Accepted: 17 May 2023 / Published online: 8 June 2023

Abstract
Point cloud completion aims to estimate the missing shape from a partial point cloud. Existing encoder-decoder based generative models usually reconstruct the complete point cloud from the learned distribution of the shape prior, which may lead to distortion of geometric details (such as sharp structures and structures without smooth surfaces) due to the information loss of the latent space embedding. To address this problem, we formulate point cloud completion as a geometric transformation problem and propose a simple yet effective geometric transformation network (GTNet). It exploits the repetitive geometric structures in common 3D objects to recover the complete shapes, which contains three sub-networks: geometric patch network, structure transformation network, and detail refinement network. Specifically, the geometric patch network iteratively discovers repetitive geometric structures that are related or similar to the missing parts. Then, the structure transformation network uses the discovered geometric structures to complete the corresponding missing parts by learning their spatial transformations such as symmetry, rotation, translation, and uniform scaling. Finally, the detail refinement network performs global optimization to eliminate unnatural structures. Extensive experiments demonstrate that the proposed method outperforms the state-of-the-art methods on the Shape-Net55-34, MVP, PCN, and KITTI datasets. Models and code will be available at https://github.com/ivislabhit/GTNet.

Keywords Point clouds · 3D shape completion · Repetitive geometric structures · Geometric transformation network

1 Introduction

With the rapid development of 3D acquisition technologies, point clouds as the preferred representations for describing 3D shapes of objects have drawn increasing attention (Qi et al., 2017a; Fei et al., 2022a; Sipiran et al., 2022). However, due to the limitations of the limited sensor resolution and self-occlusion, the raw point clouds generated by these devices are highly sparse, noisy, and severely incomplete, which leads to the loss of geometric and semantic information. Therefore, recovering complete point clouds from the incomplete raw data is important in downstream 3D computer vision applications such as robotics, autonomous driving, and 3D reconstruction (Stutz & Geiger, 2020; Sipiran et al., 2022; Wang et al., 2022b).

Due to the irregularity and disorder of point clouds, convolution can not be directly applied to them. Mainstream point cloud completion methods use the Multi-Layer Perceptrons (MLPs) to process point clouds. Inspired by the learning architecture of PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), previous works (Yang et al., 2018; Groueix et al., 2018; Yuan et al., 2018; Liu et al.,

✉ Liqiang Nie
nieliqiang@gmail.com

Shengping Zhang
s.zhang@hit.edu.cn

Xianzhu Liu
xianzhu.liu@outlook.com

Haozhe Xie
haozhexie@tencent.com

Huiyu Zhou
hz143@leicester.ac.uk

Dacheng Tao
dacheng.tao@sydney.edu.au

Xuelong Li
li@nwpu.edu.cn

1 Harbin Institute of Technology, Weihai, China

2 Tencent AI Lab, Shenzhen, China

3 Harbin Institute of Technology, Shenzhen, China

4 University of Leicester, Leicester, UK

5 The University of Sydney, Camperdown, Australia

6 Northwestern Polytechnical University, Xi'an, China

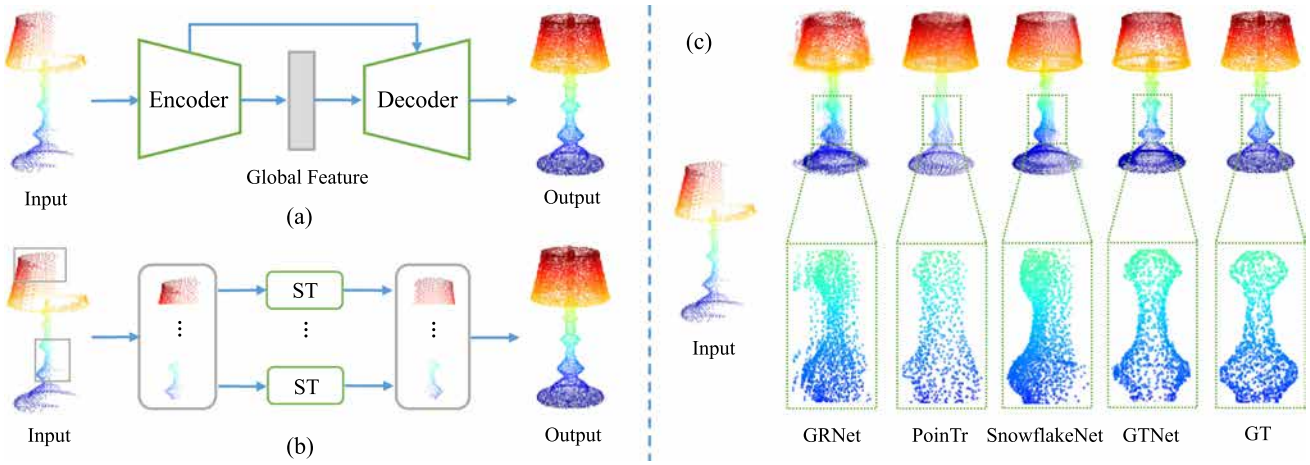

Fig. 1 **a** Most existing methods employ an encoder-decoder framework to reconstruct the complete point cloud from the learned distribution of the shape prior. **b** The proposed GTNet exploits repetitive geometric structures to recover missing parts from patch-to-patch in a Spatial Transformation (ST) manner. **c** Qualitative comparison results show that GTNet (Ours) preserves existing structures and recovers better geometric details than other methods. The striking characteristic of our network is that it does not require a complex encoder-decoder and recovers more realistic geometric details

2020; Huang et al., 2020) adopt a PointNet-based encoder to extract global features from incomplete point clouds and use a decoder to predict missing parts. However, max-pooling in the PointNet-based encoder ignores the structural information of local regions, which leads to the loss of fine-grained structures. To preserve local region features, recent methods (e.g., SANet (Wen et al., 2020), CRN (Wang et al., 2020b), and SpareNet (Xie et al., 2021)) combine global features with local details to reconstruct complete point clouds. They regard point cloud completion as a point cloud generation task and exploit an encoder-decoder framework to reconstruct the complete point cloud from the learned distribution of the shape prior, which, however, may lead to the distortion of geometric details such as sharp structures and structures without smooth surfaces. The main reasons are: (1) the embedding in latent space inevitably causes information loss due to the compression in the encoder and (2) it is difficult for generative models to generate detailed geometric structures due to the disorder and discreteness of point clouds. Furthermore, they do not consider exploiting the ubiquitous repetitive geometric structures in common 3D objects, which play an important role in fields such as biology, architecture, and art (Niloy J. et al., 2006; Mark et al., 2008). As illustrated in Fig. 1, the right part of the lamp is missing, which can be easily recovered by rotating around the lamp pole. Several works (Thrun & Wegbreit, 2005; Sung et al., 2015; Schiebener et al., 2016) attempt to discover the symmetrical geometric structures in objects to complete the missing parts. However, they can not recover the missing parts of objects that are asymmetrical. Besides symmetry, the geometric structures with rotation, translation, uniform scaling, or their combinations should also be considered because they occur frequently in common 3D objects.

To address these problems, we abandon the encoder-decoder framework and explore how to use the repetitive geometric structures in common 3D objects to recover missing geometric structures in a spatial transformation manner. To this aim, we formulate point cloud completion as a geometric transformation problem from a new perspective and propose a new Geometric Transformation Network (GTNet), which is composed of three sub-networks: geometric patch network, structure transformation network, and detail refinement network. GTNet is trained to progressively discover and transform the repetitive geometric structures from input partial point clouds into the corresponding missing parts in a patch-to-patch fashion. In particular, the geometric patch network iteratively discovers repetitive geometric structures (i.e., geometric patches) that are related or similar to the missing parts through a learnable scorer. The scorer learns higher scores for points belonging to a repetitive geometric structure and lower scores for other points each time. The structure transformation network is then used to learn a spatial transformation matrix for each repetitive geometric structure, which completes the missing parts by spatial transformations, including symmetry, rotation, translation, uniform scaling, and their combinations. The detail refinement network is finally used for global optimization, which completes the missing parts without corresponding repetitive geometric structures in the partial point clouds. Compared to existing encoder-decoder based generative models, the proposed method not only preserves existing local geometric details but also recovers more realistic and fine-grained missing parts.

In summary, our main contributions are summarized as follows:

- We formulate point cloud completion as a geometric transformation problem and propose to exploit the repetitive geometric structures to complete the missing parts in a spatial transformation manner.
- We present a geometric patch network to efficiently discover repetitive geometric structures with symmetry, rotation, translation, uniform scaling, and their combinations, which are widely existing in common 3D objects.
- Extensive experiments demonstrate that the proposed GTNet outperforms the state-of-the-art methods on the ShapeNet55-34, MVP, PCN, and KITTI datasets.

2 Related Work

In recent years, deep learning-based point cloud completion methods have been rapidly developed. In this section, we briefly review three mainstream approaches: Voxel-based, GAN-based, and Point-based. Comprehensive reviews of point cloud completion methods can be found in Fei et al. (2022a).

2.1 Voxel-Based Learning Methods

Inspired by the successful application of CNN in image restoration (Fu et al., 2017; Zheng et al., 2021; Zhang et al., 2022), super-resolution (Buades et al., 2019; Zhang et al., 2020b; Yang et al., 2022), and 3D reconstruction (Xie et al., 2019, 2020a; Wang et al., 2019) research, the intuitive idea of 3D shape completion is to divide 3D space into regular voxels and apply CNN to 3D space. Wu et al. (2015) first propose to represent 3D shapes as probability distributions of binary variables on 3D voxel grids and then use 3D CNN to generate complete 3D shapes. Then, Han et al. (2017) uses a 3D decoder to progressively generate high-resolution, complete 3D shapes with a joint global structure inference and local geometric refinement strategy. GRNet (Xie et al., 2020b) reconstructs complete point clouds from coarse to fine, which first uses 3D CNN to predict the coarse complete shapes and then converts the output voxels into point clouds for further refinement. However, the memory and computational cost of voxel-based networks increases rapidly with the increase of spatial resolution, which is cubic to the resolution of 3D voxels. Therefore, it is difficult to generate high-resolution complete point clouds (Dai et al., 2017; Stutz & Geiger, 2018). To solve this problem, some works apply different partitioning strategies (Klokov & Lempitsky, 2017; Riegler et al., 2017; Tatarchenko et al., 2017) and sparse representation (Graham et al., 2018; Su et al., 2018; Wang et al., 2018). However, the quantization effect inevitably leads to the loss of information, which is not suitable for fine-grained generation tasks.

2.2 GAN-Based Learning Methods

Since the GAN networks (Goodfellow et al., 2014; Arjovsky et al., 2017; Chrysos et al., 2020) can avoid complex optimization processes and speed up predictions, they are very suitable for generating tasks on point clouds (Li et al., 2019; Shu et al., 2019; Yan et al., 2020; Chen et al., 2020; Hui et al., 2020). L-GAN (Achlioptas et al., 2018) is the first deep generation model for point clouds. RL-GAN-Net (Sarmad et al., 2019) is the first GAN network based on reinforcement learning agent control, which can eliminate the costly and complex optimization process. Wu et al. (2020) further propose the first multi-modal shape completion method, which associates an incomplete shape with multiple different complete shapes via conditional generation modeling. To improve the network comprehension ability and the completion accuracy, Cycle4Completion (Wen et al., 2021a) establishes the geometric correspondence between incomplete and complete point clouds in two directions: incomplete-cycle and complete-cycle. These methods generally encode the input point cloud into a global feature vector through an encoder network. However, the global features extracted from the incomplete point clouds necessarily lose complete shape information and discard many local geometric details, which makes it impossible to generate fine-grained complete point clouds. Furthermore, some point-based works use adversarial loss to improve the consistency and reality of the reconstructed point clouds with the ground truth point clouds. For example, PF-Net (Huang et al., 2020), CRN (Wang et al., 2020b), and SpareNet (Xie et al., 2021) obtain more realistic local geometry structures by training their networks with a combination of completion loss and adversarial loss. These works demonstrate the potential of incorporating adversarial loss as an auxiliary loss function in training models, leading to more realistic local geometry structures.

2.3 Point-Based Learning Methods

The point-based learning networks can reduce the time overhead of conversion between point clouds and other data formats (such as voxels) while avoiding the loss of detailed information due to data conversion. Pioneered by PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), PCN (Yuan et al., 2018) is the first point-based point cloud completion network based on the encoder-decoder architecture and folding mechanism (Yang et al., 2018). Further, PF-Net (Huang et al., 2020) extracts the multi-scale features of point clouds and uses a multi-stage strategy to generate the missing points. To solve the problem that only using global features to reconstruct the complete point clouds may lead

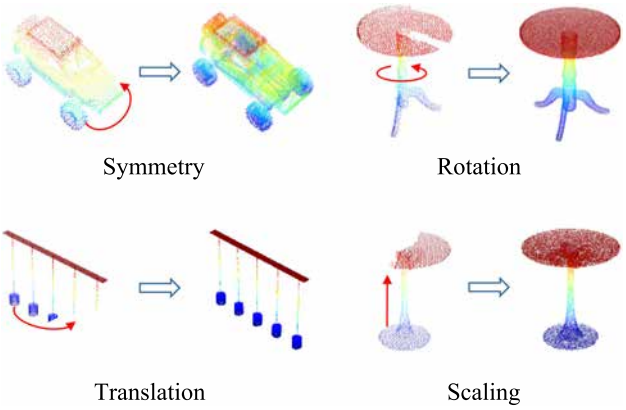

Fig. 2 Examples of repetitive geometric structures can be used to complete missing shapes in point clouds. The repetitive geometric structures discovered by our algorithm are used to complete the missing parts through spatial transformations, including symmetry, rotation, translation, scaling, and their combinations

to the loss of rich local region information, a series of methods (Wen et al., 2020; Zhang et al., 2020a; Wang et al., 2020b; Xiang et al., 2021; Xie et al., 2021; Yu et al., 2021) have been proposed. PoinTr (Yu et al., 2021) uses a transformer-based architecture with geometric perception blocks to predict the initial sketch of the missing parts, which can better learn structural knowledge and retain the fine-grained information of the point clouds. SnowflakeNet (Xiang et al., 2021) models the point cloud completion as a snowflake-like growth of points in the 3D space, which enables the network to learn the most suitable point splitting pattern for the local regions to reconstruct highly detailed geometric shapes. However, these methods mainly learn the deterministic partial-to-complete mapping, ignoring the importance of the relationship structure attributes in objects for restoring the complete shapes. Recent works (Wang et al., 2020a; Zhao et al., 2021; Pan et al., 2021; Xia et al., 2021) show that fine-grained structural details can be generated by learning shape priors from the complete point clouds, which holds exciting prospects for point cloud completion. Furthermore, some interesting works (Zhou et al., 2021; Lyu et al., 2022) analogize points in point clouds as particles in thermodynamic systems inspired by non-equilibrium thermodynamics, which achieves highly competitive performance by employing a diffusion process in point cloud generation. For example, PDR (Lyu et al., 2022) uses a diffusion model to establish a one-to-one pointwise mapping between the generated point cloud and the uniform ground truth, which accurately manipulates spatial locations of 3D points to obtain smooth surfaces and sharp details. Unlike them, we use the repetitive geometric structures in common 3D objects to recover missing geometric structures in a spatial transformation manner. It not only avoids complex encoder-decoder design but also recovers more realistic geometric details.

3 The Proposed Method

3.1 Overview

Given a partial point cloud $\mathbf{P} \in \mathbb{R}^{N_p \times 3}$, our goal is to recover a complete point cloud $\mathbf{Q} \in \mathbb{R}^{N_q \times 3}$ by means of geometric transformations, where N_p and N_q denote the numbers of points in $\mathbf{P}$ and $\mathbf{Q}$, respectively. Note that $\mathbf{P}$ is generally not a subset of $\mathbf{Q}$. As illustrated in Fig. 2, the missing shapes of the partial point clouds can be easily recovered by symmetry, rotation, translation, and scaling of their corresponding repetitive geometric structures. The overall architecture of the proposed Geometric Transformation Network (GTNet) is shown in Fig. 3, which is composed of three sub-networks: geometric patch network, structure transformation network, and detail refinement network. The details of these sub-networks are introduced as follows.

3.2 Geometric Patch Network

The Geometric Patch Network (GPN) aims to efficiently discover repetitive geometric structures without prior knowledge of their geometry or spatial location. Specifically, GPN iteratively aggregates local geometric structures that are related or similar to the missing parts as repetitive geometric structures. It consists of a feature extraction module and a patch generator module.

3.2.1 Feature Extraction

For each point $p_i = [x_i, y_i, z_i]$ in the partial point cloud $\mathbf{P}$, we first collect its K neighboring points, i.e., $\mathcal{N}_i = \{p_i^k\}_{k=1}^K$, through the K-nearest neighbor algorithm. The local geometric structure of the center point p_i is formulated as $\widetilde{\mathcal{N}}_i = \{p_i, p_i^k, p_i - p_i^k, \|p_i - p_i^k\|\}_{k=1}^K$. Then, a Local Encoder (LE) is used to extract local features $\mathbf{f}_i = \mathrm{LE}(\widetilde{\mathcal{N}}_i) \in \mathbb{R}^{1 \times d_1}$ for p_i. In this way, LE can explicitly encode local geometric structures, which ultimately facilitates GPN to efficiently discover repetitive geometric structures. The $\mathrm{LE}(\cdot)$ is implemented by two shared Multi-Layer Perceptrons (MLPs) with a max-pooling operation. Note that it can be replaced with any other well-designed better encoder. By extracting local features for each point, we obtain the local features $\mathbf{F}_{local} = \{\mathbf{f}_i\}_{i=1}^{N_p} \in \mathbb{R}^{N_p \times d_1}$ for $\mathbf{P}$. In this work, we set $d_1 = 128$.

Effective local–global features of partial point clouds are the key to efficiently discover repetitive geometric structures. To obtain effective global features, we use the self-attention mechanism to refine local features according to the global context. Let $\mathbf{Q}, \mathbf{K}$, and $\mathbf{V}$ be the query, key, and value matrices, respectively, which are generated as follows

$$(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{F}_{local} \cdot (\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v) \tag{1}$$

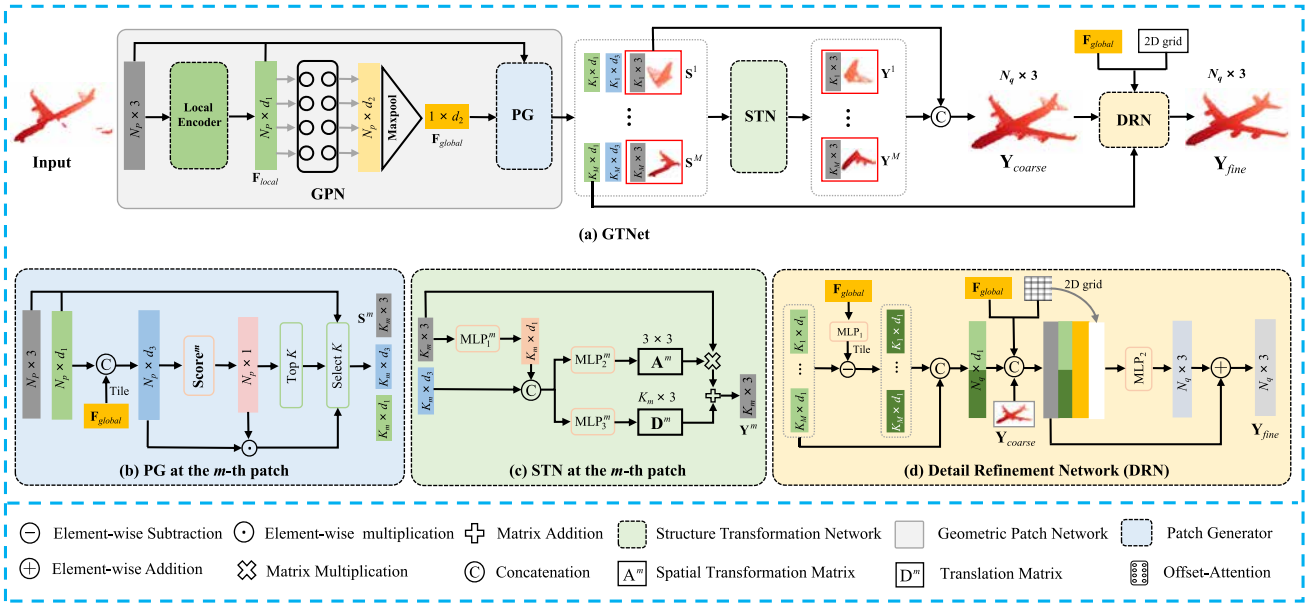

Fig. 3 Illustration of the proposed network. **a** The overall architecture of GTNet. **b** The details of PG at the m-th patch. **c** The details of STN at the m-th patch. **d** The details of DRN. Note that N_p and N_q are the number of points, d_1, d_2, and d_3 are the number of point feature dimensions that are 128, 512, and 640, respectively

where $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{d_1 \times c_1}$ and $\mathbf{W}_v \in \mathbb{R}^{d_1 \times c_2}$ are the shared learnable linear transformation matrices. To be computationally efficient, the dimensions of the query and key vectors are set to 1/4 of the value vector (i.e., $c_1 = d_1/4$, $c_2 = d_1$). Then, the attention weights $\mathbf{A}$ and self-attention output features $\mathbf{F}_{sa}$ are obtained by

$$\begin{aligned} \mathbf{A} &= \mathbf{Q} \cdot \mathbf{K}^T \\ \mathbf{F}_{sa} &= \mathbf{A} \cdot \mathbf{V} \end{aligned} \tag{2}$$

To consider the semantic information and neighbor embedding, we improve the self-attention with the offset attention (OA) proposed in PCT (Guo et al., 2021). Specifically, OA generates global features $\mathbf{F}_{oa}$ with the self-attention features $\mathbf{F}_{sa}$ and the local features $\mathbf{F}_{local}$, which can be formulated as

$$\begin{aligned} \mathbf{F}_1 &= \mathrm{OA}(\mathbf{F}_{local}) = (\mathbf{F}_{local} - \mathbf{F}_{sa}) \cdot \mathbf{W_1} + \mathbf{F}_{local} \\ \mathbf{F}_i &= \mathrm{OA}^i(\mathbf{F}_{i-1}), i = 2, 3, 4 \\ \mathbf{F}_{oa} &= \mathrm{concat}(\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4) \cdot \mathbf{W}_o \end{aligned} \tag{3}$$

where $\mathbf{W_1} \in \mathbb{R}^{d_1 \times d_1}$ and $\mathbf{W}_o \in \mathbb{R}^{4d_1 \times d_2}$ are the weights of linear layers. OA^i represents the i-th attention layer.

Finally, a point-wise max-pooling layer is applied on $\mathbf{F}_{oa}$ to extract d_2-dimensional global features $\mathbf{F}_{global} \in \mathbb{R}^{1 \times d_2}$. In this work, we set $d_2 = 512$.

3.2.2 Patch Generator

Based on the extracted local–global features, the Patch Generator (PG) is used to iteratively aggregate local geometric structures that are related or similar to the missing parts as repetitive geometric structures. In practice, since the scale of each local geometric structure is different, the number of points in each repetitive geometric structure is arbitrary. However, this is difficult and inefficient for deep networks. For simplicity and efficiency, we set the number of points in each repetitive geometric structure to K. In this way, large-scale geometric structures may be relatively sparse but can be made dense with multiple iterations. Note that a reasonable K value is critical to obtain effective repetitive geometric structures, which is discussed in Sect. 4.7.6.

In our approach, PG iteratively generates M geometric patches $\mathcal{S} = \{\mathbf{S}^m\}_{m=1}^M$ from $\mathbf{P}$ adaptively. Each patch represents a repetitive geometric structure. The value of M is proportional to the number of points in the ground truth point clouds, i.e. $M = N_q/2K$. As shown in Fig. 3b, with the local–global features $\mathbf{F}_{lg} = \mathrm{cat}(\mathbf{F}_{local}, \mathbf{F}_{global})$ as inputs (cat denotes the concatenation operation), PG at the m-th patch calculates a score for each point in $\mathbf{P}$

$$\mathbf{s}^m = \mathrm{Score}^m(\mathbf{F}_{lg}) \in \mathbb{R}^{N_p \times 1} \tag{4}$$

where $\mathrm{Score}^m(\cdot)$ is the scoring function implemented by MLPs. Since the local features of points belonging to the same geometric structure are similar, it learns higher scores

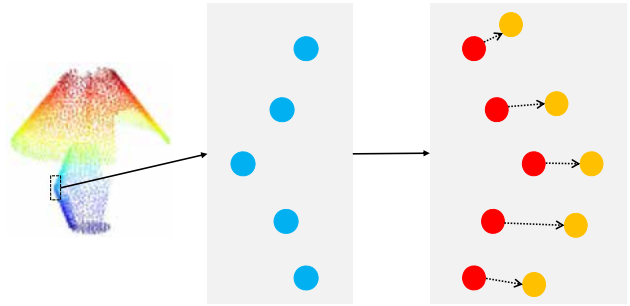

Fig. 4 Illustration of the effectiveness of $\mathbf{D}^m$. When points (blue) in the input repetitive geometric structure are deformed into target points (red) by linear spatial transformation, there are coordinate offsets from the ground truth points (yellow). $\mathbf{D}^m$ is used to correct these offsets with different orientations and magnitudes by providing an extra freedom of movement for each point (Color figure online)

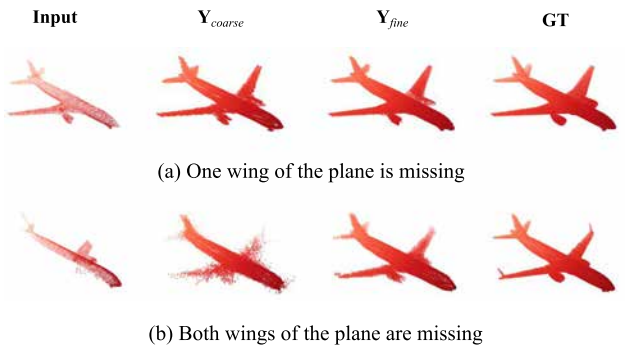

Fig. 5 Effectiveness of the detail refinement network. If the missing geometric structures can not be inferred directly from the partial point clouds, the detail refinement network is used for global optimization to eliminate the unnatural geometric structures generated by the structure transformation network

for points belonging to a repetitive geometric structure in $\mathbf{P}$ and lower scores for other points. Specifically, points with top-K ($K < N_p$) scores are selected to form a geometric patch $\mathbf{S}^m \in \mathbb{R}^{K\times 3}$, which can be formulated as

$$\begin{aligned} \mathbf{idx}^m &= \arg\ \mathrm{top}_K(\mathbf{s}^m) \\ \mathbf{S}^m &= \mathbf{P}[\mathbf{idx}^m] \\ \mathbf{F}^m_{local} &= \mathbf{F}_{local}[\mathbf{idx}^m] \end{aligned} \tag{5}$$

where $\mathbf{idx}^m$ is the index vector of the top-K points and $\mathbf{F}^m_{local}$ are the corresponding local features of $\mathbf{S}^m$.

To make the generation process of the geometric patches differentiable, we use the sigmoid function after $\mathrm{Score}^m(\cdot)$ to generate local–global features $\mathbf{F}^m_{lg}$ of the m-th patch for subsequent processing

$$\mathbf{F}^m_{lg} = \mathbf{F}_{lg}[\mathbf{idx}^m] \odot \mathrm{sigmoid}(\mathbf{s}^m[\mathbf{idx}^m]) \tag{6}$$

where $\mathbf{s}^m[\mathbf{idx}^m]$ represents the highest K scores and $\odot$ represents element-wise multiplication.

3.3 Structure Transformation Network

The Structure Transformation Network (STN) is designed to perform spatial transformations, including symmetry, rotation, translation, uniform scaling, and their combinations, on the discovered repetitive geometric structures to recover the corresponding missing structures. An intuitive idea for modeling these spatial transformations is to use 3D affine transformations, which can be formulated as

$$\mathbf{Y}^m = \mathbf{S}^m\mathbf{A}^m + \mathbf{b}^m \tag{7}$$

where $\mathbf{A}^m \in \mathbb{R}^{3\times 3}$ and $\mathbf{b}^m \in \mathbb{R}^{1\times 3}$ are the affine transformation matrix and the translation vector, respectively. $\mathbf{Y}^m$ is the corresponding missing structure.

However, the affine transformation is linear and the translation vector $\mathbf{b}^m$ is the same for all points, which makes it impossible to change the local neighborhood of the point clouds. As shown in Fig. 4, there are coordinate offsets between $\mathbf{Y}^m$ and the corresponding ground truth due to the disorder and discreteness of point clouds. However, $\mathbf{b}^m$ is unable to correct these offsets with different directions and magnitudes. To address this problem, the proposed STN learns a translation matrix $\mathbf{D}^m$ to replace the translation vector $\mathbf{b}^m$, which provides an extra freedom of movement for each point. In this way, a simple local nonlinear deformable spatial transformation for $\mathbf{S}^m$ can be formulated as

$$\mathbf{Y}^m = \mathbf{S}^m\mathbf{A}^m + \mathbf{D}^m \tag{8}$$

where $\mathbf{D}^m \in \mathbb{R}^{K_m\times 3}$. As shown in Fig. 3c, three MLPs are used to learn $\mathbf{A}^m$ and $\mathbf{D}^m$ from the local–global features $\mathbf{F}^m_{lg}$ and patch $\mathbf{S}^m$ in an end-to-end manner. The first one is used as position encoding to extract local features of the geometric patch $\mathbf{S}^m$. Then, the remaining two use the local features and $\mathbf{F}^m_{lg}$ as inputs to predict $\mathbf{A}^m$ and $\mathbf{D}^m$, respectively.

After the spatial transformation matrices are learned for all M geometric patches, STN follows Eq. 8 to complete the missing parts and concatenate them as the coarse complete point cloud $\mathbf{Y}_{coarse}$.

3.4 Detail Refinement Network

The Detailed Refinement Network (DRN) is designed to handle the case that no corresponding repetitive geometric structures are found in the partial point clouds. As shown in Fig. 5, STN generates unexpected completion results when both wings of the plane are missing, which can be improved by the proposed DRN. In addition, it can also eliminate the unnatural structures caused by the overlaps among geometric patches.

To correct the unexpected structures, DRN learns to move points from the wrongly recovered structures to the target structures under the guidance of global features. Through the point displacement prediction process, it learns a point-level mapping between the coarse output and the target. As shown in Fig. 3d, the outputs of DRN are the coordinate offsets of each point between $\mathbf{Y}_{coarse}$ and the ground truth. Besides using global features of the partial point clouds, we also introduce local features from $\mathbf{Y}_{coarse}$, which preserves the local details of the point clouds. Specifically, the local features $\mathbf{F}_{res}^{m}$ of $\mathbf{Y}^{m}$ can be computed as

$$\mathbf{F}_{res}^{m} = \mathrm{MLP}_1(\mathbf{F}_{global}) - \mathbf{F}_{local}^{m} \tag{9}$$

where MLP_1 is the first MLPs layer in DRN. The local features are concatenated to generate $\mathbf{F}_{coarse} = \left\{\mathrm{cat}(\mathbf{F}_{res}^{m}, \mathbf{F}_{local}^{m})\right\}_{m=1}^{M}$. Inspired by FoldingNet (Yang et al., 2018) and AtlasNet (Groueix et al., 2018), we introduce a 2D grid to increase the correlation between the points, where each point in $\mathbf{Y}_{coarse}$ is connected with the 2D grid coordinate. The final point cloud $\mathbf{Y}_{fine}$ can be obtained by

$$\begin{aligned}\mathbf{Y}_{fine} = &\ \mathrm{MLP}_2(\mathrm{cat}(\mathbf{Y}_{coarse}, \mathbf{F}_{2D}, \mathbf{F}_{coarse}, \mathbf{F}_{global})) \\ &+ \mathbf{Y}_{coarse}\end{aligned} \tag{10}$$

where MLP_2 is the second MLPs layer in DRN. $\mathbf{F}_{2D} \in \mathbb{R}^{N_q \times 2}$ are the 2D grid coordinates.

3.5 Loss Function

We use the symmetric Chamfer Distance (CD) as the loss function, which measures the average distance between the spatial nearest neighbors of two point clouds. Given two point clouds $\mathbf{P}$ and $\mathbf{Q}$, CD can be formulated as

$$\begin{aligned}L_{CD}(\mathbf{P}, \mathbf{Q}) = &\frac{1}{|\mathbf{P}|}\sum_{x\in\mathbf{P}} \min_{y\in\mathbf{Q}} \|x - y\|_2^2 \\ &+ \frac{1}{|\mathbf{Q}|}\sum_{y\in\mathbf{Q}} \min_{x\in\mathbf{P}} \|x - y\|_2^2\end{aligned} \tag{11}$$

Our total loss function consists of two parts. The first part calculates the distance between the coarse output $\mathbf{Y}_{coarse}$ and the ground truth $\mathbf{Y}_{gt}$. The second part calculates the distance between the refined output $\mathbf{Y}_{fine}$ and the $\mathbf{Y}_{gt}$

$$Loss = L_{CD}(\mathbf{Y}_{coarse}, \mathbf{Y}_{gt}) + \alpha L_{CD}(\mathbf{Y}_{fine}, \mathbf{Y}_{gt}) \tag{12}$$

where α is a hyperparameter, which controls the weights of the two parts.

4 Experiments

4.1 Datasets

ShapeNet55-34 The ShapeNet55-34 dataset (Yu et al., 2021) is created from the ShapeNet dataset (Chang et al., 2015), which includes two benchmarks: ShapeNet-55 and ShapeNet-34. The ShapeNet55 benchmark contains 52,470 objects from 55 categories, which aims to comprehensively evaluate the performance of networks on a more diverse dataset. We adopt the same split strategy as PoinTr (Yu et al., 2021), where 80% objects are randomly selected to form the training set and the remaining objects are used as the test set. The ShapeNet-34 benchmark is obtained by dividing the original ShapeNet-55 benchmark into 34 seen categories and 21 unseen categories, which aims to evaluate the generalization ability of networks on unseen categories. The training set and the test set for the seen and unseen categories contain 46,765, 3400, and 2305 objects, respectively. In both benchmarks, the ground truth point clouds and partial point clouds contain 8192 and 2048 points, respectively.

MVP The MVP dataset (Pan et al., 2021) is a high-quality multi-view partial point cloud dataset, which consists of more than 100,000 point clouds from 16 categories: airplane, cabinet, car, chair, lamp, sofa, table, watercraft, bed, bench, bookshelf, bus, guitar, motorbike, pistol, and skateboard. Following VRCNet (Pan et al., 2021), we evaluate the performance of point cloud completion at different resolutions, including 2048(1×), 4096(2×), 8192(4×), and 16,384(8×). The ground truth point clouds are Poisson disk sampled from the surface of CAD models. The partial point clouds are rendered from 26 uniformly distributed camera poses for each 3D CAD model and contain 2048 points. For a fair comparison, we use the same train and test splits as in VRCNet (Pan et al., 2021).

PCN The PCN dataset (Yuan et al., 2018) is a subset derived from the ShapeNet dataset (Chang et al., 2015), which is a widely used dataset for point cloud completion. It contains 30,974 CAD models from 8 categories, which are the same as the first 8 categories of the MVP dataset. The training set contains 28,974 CAD models, each of which contains a complete point cloud and corresponding 8 partial point clouds. The ground truth point clouds are uniformly sampled from the surface of CAD models and contain 16,384 points. The partial point clouds are generated by back-projecting 2.5D depth images into 3D from 8 random viewpoints, where the number of points in each point cloud is not fixed. We use the same split settings as in PCN (Yuan et al., 2018) for a fair comparison.

KITTI The KITTI dataset (Geiger et al., 2013) is composed of a sequence of real-world Velodyne LiDAR scans, which is a widely used dataset for completing sparse point clouds of cars in real-world environments. It contains 2401 par-

Table 1 Quantitative comparison results on the ShapeNet-55 benchmark in terms of per-point L2 Chamfer Distance $\times 10^3$ and F-Score@1%

Method	Table	Chair	Airplane	Car	Sofa	Bird house	Bag	Remote	Key board	Rocket	CD-S	CD-M	CD-H	CD-Avg	F1
FoldingNet (Yang et al., 2018)	2.53	2.81	1.43	1.98	2.48	4.71	2.79	1.44	1.24	1.48	2.67	2.66	4.05	3.12	0.082
TopNet (Tchapmi et al., 2019)	2.21	2.53	1.14	2.18	2.36	4.83	2.93	1.49	0.95	1.32	2.26	2.16	4.3	2.91	0.126
PCN (Yuan et al., 2018)	2.13	2.29	1.02	1.85	2.06	4.50	2.86	1.33	0.89	1.32	1.94	1.96	4.08	2.66	0.133
PFNet (Huang et al., 2020)	3.95	4.24	1.81	2.53	3.34	6.21	4.96	2.91	1.29	2.36	3.83	3.87	7.97	5.22	0.339
GRNet (Xie et al., 2020b)	1.63	1.88	1.02	1.64	1.72	2.97	2.06	1.09	0.89	1.03	1.35	1.71	2.85	1.97	0.238
PoinTr (Yu et al., 2021)	0.81	0.95	0.44	0.91	0.79	1.86	0.93	0.53	0.38	0.57	0.58	0.88	1.79	1.09	0.464
GTNet	**0.71**	**0.88**	**0.42**	**0.85**	**0.75**	**1.61**	**0.87**	**0.48**	**0.33**	**0.50**	**0.45**	**0.66**	**1.30**	**0.80**	**0.543**

We report the detailed results of 10 categories and the average results of 55 categories for three difficulty degrees: Simple (CD-S), Moderate (CD-M), and Hard (CD-H). The best values are highlighted in bold

Table 2 Quantitative comparison results on the ShapeNet-34 benchmark in terms of per-point L2 Chamfer Distance $\times 10^3$ and F-Score@1%

Method	34 seen categories					21 unseen categories				
	CD-S	CD-M	CD-H	CD-Avg	F1	CD-S	CD-M	CD-H	CD-Avg	F1
FoldingNet (Yang et al., 2018)	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
TopNet (Tchapmi et al., 2019)	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PCN (Yuan et al., 2018)	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
PFNet (Huang et al., 2020)	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
GRNet (Xie et al., 2020b)	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
PoinTr (Yu et al., 2021)	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
GTNet	**0.51**	**0.73**	**1.40**	**0.88**	**0.511**	**0.78**	**1.22**	**2.56**	**1.52**	**0.467**

We report the average results of 34 seen categories and 21 unseen categories for three difficulty degrees: Simple (CD-S), Moderate (CD-M), and Hard (CD-H). The best values are highlighted in bold

tial car point clouds, which are collected from 98 real cars under 425 different frames according to the coordinates of 3D bounding boxes. Unlike partial point clouds generated by back-projection of 2.5D images in synthetic datasets, the scanned data from LiDAR scans are significantly sparser, with an average of only 440 points and some containing fewer than 10 points. Note that the partial point clouds in KITTI do not have complete point clouds as ground truth.

4.2 Implementation Details

We implement our network using PyTorch (Paszke et al., 2019). The hyperparameter α is set to 1. We train the network using the Adam optimizer (Kingma & Ba, 2015) with a batch size of 32 on two NVIDIA GeForce RTX 3090 GPUs. The initial learning rate is set to $1e^{-4}$ and decayed by 0.7 after 40 epochs. The optimization is set to stop after 200 epochs.

4.3 Evaluation on the ShapeNet55-34 dataset

To compare the performance of GTNet with other state-of-the-art methods, including FoldingNet (Yang et al., 2018), TopNet (Tchapmi et al., 2019), PCN (Yuan et al., 2018), PFNet (Huang et al., 2020), GRNet (Xie et al., 2020b), and PoinTr (Yu et al., 2021), we conduct experiments on the ShapeNet55-34 dataset at three difficulty degrees: simple, moderate, and hard. Tables 1 and 2 show the quantitative comparison results on the ShapeNet-55 and ShapeNet-34 benchmark, respectively. It is observed from Table 1 that compared to PoinTr (Yu et al., 2021) on the ShapeNet55 benchmark, GTNet achieves 0.13, 0.22, and 0.49 improvements in the average chamfer distance values $\times 10^3$ under simple, moderate, and hard settings, respectively. Table 2 further shows that GTNet outperforms all other methods both in 34 seen categories and 21 unseen categories on the ShapeNet-34 benchmark. Especially, compared to the second-ranked method PoinTr (Yu et al., 2021), GTNet improves the average chamfer distance values and F-Score@1% by 25.9% and 21.6% on 21 unseen categories, respectively. It demonstrates the excellent generalization ability of our method on unseen categories.

Furthermore, Fig. 6 shows the qualitative comparison results on the ShapeNet55-34 dataset. Compared to the state-of-the-art methods, GTNet recovers more precise detailed structures based on the existing geometric structures. For example, in the eighth row of Fig. 6, FoldingNet (Yang et al., 2018) and PCN (Yuan et al., 2018) fail to complete the missing parts. GRNet (Xie et al., 2020b) and PoinTr (Yu et al., 2021) are able to recover coarse shapes but fail to generate satisfactory results. In contrast, GTNet can infer the missing parts based on the existing geometric structures and recover the detailed structures.

In a nutshell, the proposed GTNet outperforms the state-of-the-art networks on the ShapeNet55-34 dataset in terms of both completion and generalization abilities. The reason why GTNet significantly outperforms other methods is that it can exploit the repetitive geometric structures in the partial point clouds. As shown in Fig. 6, we can find that GTNet recovers more realistic and fine-grained missing shapes when there are corresponding repetitive geometric structures in the partial point clouds.

4.4 Evaluation on the MVP Dataset

The proposed GTNet is then compared to several state-of-the-art methods on the MVP dataset. Specifically, Tables 3 and 4 show the average chamfer distance values and F-Score@1% of all existing competitive methods on the MVP test set at 16384(8×) resolution. We can observe that GTNet significantly outperforms all other methods on both evaluation metrics. Especially, compared to the second-ranked VRCNet (Pan et al., 2021), GTNet decreases the average chamfer distance values with a margin of 28.4%. Table 5 further demonstrates that GTNet outperforms all existing competitive methods at various resolutions, including 2048(1×), 4096(2×), 8192(4×), and 16384(8×). Note that all other

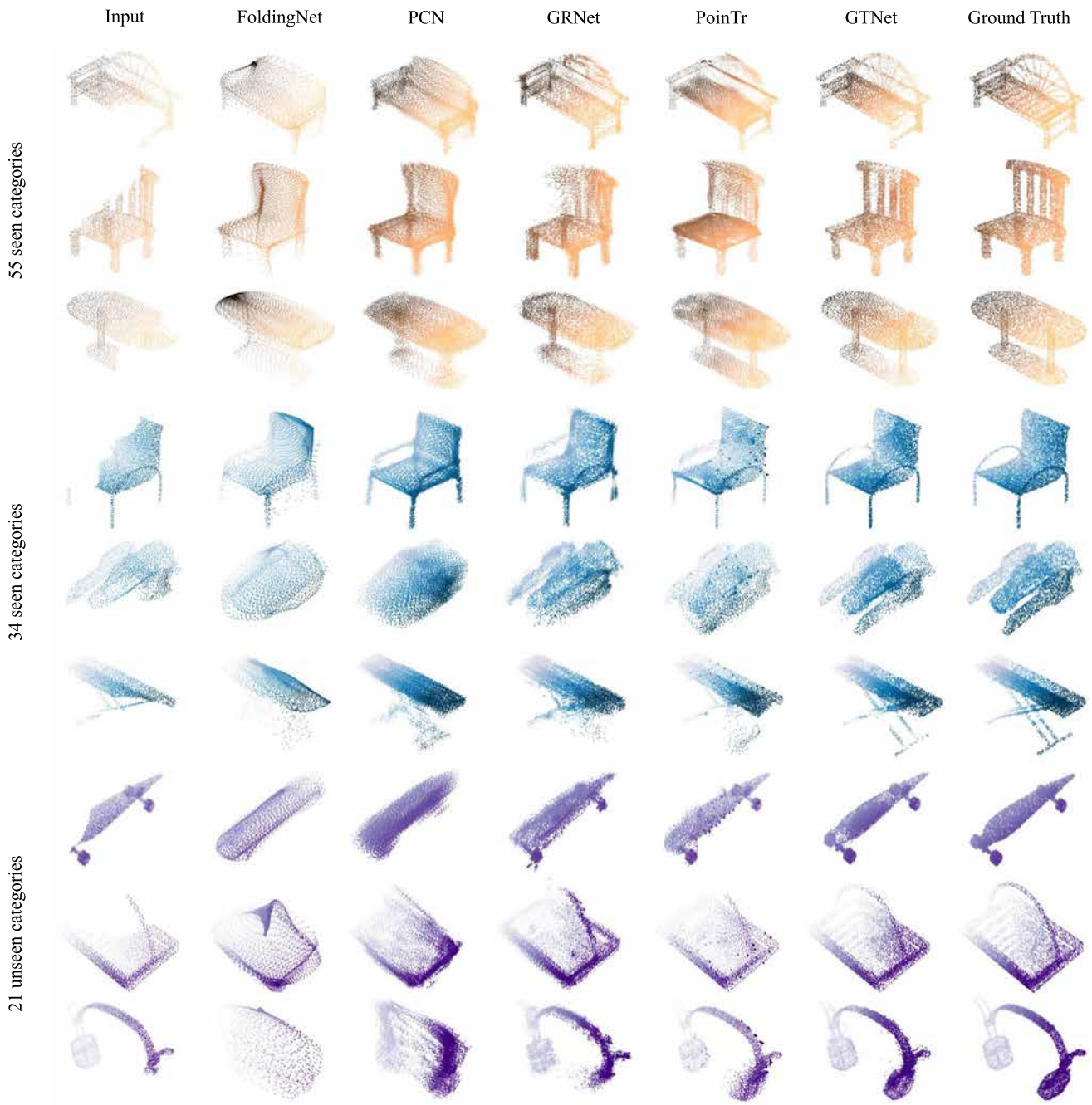

Fig. 6 Qualitative comparison results on the ShapeNet-55 and ShapeNet-34 benchmarks

methods mentioned in Tables 3, 4, and 5 are typical generative models based on the encoder-decoder framework. The significant performance improvements achieved by GTNet demonstrate the effectiveness of formulating point cloud completion as a geometric transformation problem rather than a generation problem.

In Fig. 7, we further show the qualitative comparison results on the MVP dataset, which indicates that the proposed GTNet recovers better structure details of objects than all other methods. For example, for the chair in the first column, the encoder-decoder based methods described above fail to reconstruct fine-grained local geometric structures and generate a noisy output. In contrast, GTNet recovers the missing parts by symmetry based on the existing chair legs. Compared to existing generative models, the proposed geometric transformation method can not only preserve existing structural details but also recover more realistic and fine-grained missing parts.

Table 3 Quantitative comparison results on the MVP dataset (16,384 points) in terms of per-point L2 Chamfer distance $\times 10^4$

Method	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Bed	Bench	Bookshelf	Bus	Guitar	Motorbike	Pistol	Skateboard	Avg
TopNet (Tchapmi et al., 2019)	2.72	4.25	3.40	7.95	17.01	6.04	7.42	6.04	11.60	5.62	8.22	2.37	1.33	3.90	3.97	2.09	6.36
PCN (Yuan et al., 2018)	2.95	4.13	3.04	7.07	14.93	5.56	7.06	6.08	12.72	5.73	6.91	2.46	1.02	3.53	3.28	2.99	6.02
MSN (Liu et al., 2020)	2.07	3.82	2.76	6.21	12.72	4.74	5.32	4.80	9.93	3.89	5.85	2.12	0.69	2.48	2.91	1.58	4.90
CRN (Wang et al., 2020b)	1.59	3.64	2.60	5.24	9.02	4.42	5.45	4.26	9.56	3.67	5.34	2.23	0.79	2.23	2.86	2.13	4.30
vGRNet (Xie et al., 2020b)	1.61	4.66	3.10	4.72	5.66	4.61	4.85	3.53	7.82	2.96	4.58	2.97	1.28	2.24	2.11	1.61	3.87
NSFA (Zhang et al., 2020a)	1.51	4.24	2.75	4.68	6.04	4.29	4.84	3.02	7.93	3.87	5.99	2.21	0.78	1.73	2.04	2.14	3.77
ECG (Pan, 2020)	1.41	3.44	2.36	4.58	6.95	3.81	4.27	3.38	7.46	3.10	4.82	1.99	0.59	2.05	2.31	1.66	3.58
VRCNet (Pan et al., 2021)	1.15	3.20	**2.14**	3.58	5.57	3.58	4.17	2.47	6.90	2.76	3.45	1.78	0.59	1.52	1.83	1.57	3.06
GTNet	**0.64**	**3.08**	2.15	**2.73**	**2.46**	**2.90**	mathbf2.72	**1.85**	**4.24**	**1.78**	**3.09**	**1.70**	**0.35**	**1.36**	**1.10**	**0.69**	**2.19**

The best values are highlighted in bold

Table 4 Quantitative comparison results on the MVP dataset (16,384 points) in terms of F-Score@1%

Method	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Bed	Bench	Bookshelf	Bus	Guitar	Motorbike	Pistol	Skateboard	Avg
TopNet (Tchapmi et al., 2019)	0.789	0.621	0.612	0.443	0.387	0.506	0.639	0.609	0.405	0.680	0.524	0.766	0.868	0.619	0.726	0.837	0.601
PCN (Yuan et al., 2018)	0.816	0.614	0.686	0.517	0.455	0.552	0.646	0.628	0.452	0.694	0.546	0.779	0.906	0.665	0.774	0.861	0.638
MSN (Liu et al., 2020)	0.879	0.692	0.693	0.599	0.604	0.627	0.730	0.696	0.569	0.797	0.637	0.806	0.935	0.728	0.809	0.885	0.710
CRN (Wang et al., 2020b)	0.898	0.688	0.725	0.670	0.681	0.641	0.748	0.742	0.600	0.797	0.659	0.802	0.931	0.772	0.843	0.902	0.740
GRNet (Xie et al., 2020b)	0.853	0.578	0.646	0.635	0.710	0.580	0.690	0.723	0.586	0.765	0.635	0.682	0.865	0.736	0.787	0.850	0.692
NSFA (Zhang et al., 2020a)	0.903	0.694	0.721	0.737	0.783	0.705	0.817	0.799	0.687	0.845	0.747	0.815	0.932	0.815	0.858	0.894	0.783
ECG (Pan, 2020)	0.906	0.680	0.716	0.683	0.734	0.651	0.766	0.753	0.640	0.822	0.706	0.804	0.945	0.780	0.835	0.897	0.753
VRCNet (Pan et al., 2021)	0.928	0.721	0.756	0.743	0.789	0.696	0.813	0.800	0.674	0.863	0.755	0.832	0.960	0.834	0.887	0.930	0.796
GTNet	**0.942**	**0.744**	**0.757**	**0.782**	**0.839**	**0.749**	**0.849**	**0.818**	**0.722**	**0.894**	**0.786**	**0.847**	**0.979**	**0.835**	**0.893**	**0.950**	**0.823**

The best values are highlighted in bold

Table 5 Quantitative comparison results at various resolutions on the MVP dataset in terms of per-point L2 Chamfer distance $\times 10^4$

Points	2048 (1×)	4096 (2×)	8192 (4×)	16384 (8×)
TopNet (Tchapmi et al., 2019)	10.11	8.20	7.00	6.36
PCN (Yuan et al., 2018)	9.77	7.96	6.99	6.02
MSN (Liu et al., 2020)	7.90	6.17	5.42	4.90
CRN (Wang et al., 2020b)	7.25	5.83	4.90	4.30
ECG (Pan, 2020)	6.64	5.41	4.18	3.58
VRCNet (Pan et al., 2021)	5.96	4.70	3.64	3.06
PDR (Lyu et al., 2022)	**5.66**	4.26	3.35	2.61
GTNet	5.76	**4.18**	**3.05**	**2.19**

We report the mean CD for all categories. The best results are highlighted in bold

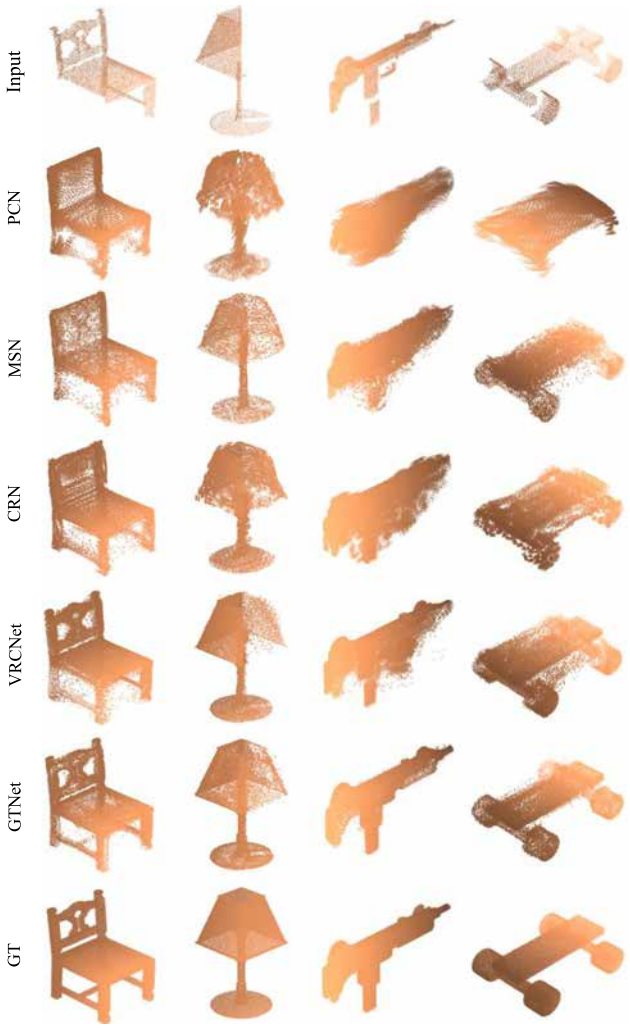

Fig. 7 Qualitative completion results on the MVP dataset. GT stands for the ground truth of the 3D object

4.5 Evaluation on the PCN Dataset

We further evaluate the proposed method against other state-of-the-art methods on the PCN dataset. The quantitative and qualitative comparison results are shown in Table 6 and Fig. 8, respectively. Table 6 shows that the proposed GTNet achieves the lowest average chamfer distance value. The visual comparison results show that GTNet can recover missing parts with better shape quality. In particular, it can preserve and recover more geometric details in small-scale structures.

Table 6 also shows that the proposed GTNet does not completely outperform PoinTr (Yu et al., 2021) and SnowflakeNet (Xiang et al., 2021). The reason is that the performance of GTNet is affected by the number of repetitive geometric structures in the dataset. We conduct an analysis on the number of partial point clouds containing repetitive geometric structures with symmetry, rotation, translation, and uniform scaling in the PCN datasets. The results indicate that only 59.2% of the partial point clouds contain repetitive geometric structures, which is significantly lower than the ShapeNet55 (76.3%) and MVP (80.8%) datasets. It makes the proposed structure transformation network in GTNet unable to recover the missing shapes by the geometric transformation in 40.8% cases. As mentioned in Sect. 3.4, the proposed Detailed Refinement Network (DRN) is used to eliminate the unexpected structures and generate missing parts based on the local–global features extracted from partial point clouds. In this case, DRN can be seen as a simple decoder and its performance is obviously weaker than the well-designed complex decoder in PoinTr and SnowflakeNet. Consequently, GTNet only achieves slight improvements in the completion performance since the number of repetitive geometric structures in the dataset is low.

4.6 Evaluation on the KITTI Dataset

Finally, we evaluate the proposed method against other state-of-the-art methods on the KITTI dataset for completing sparse point clouds of cars in real-world environments. For a fair comparison, we follow the experimental settings of previous methods (Xie et al., 2020b; Yu et al., 2021; Xiang et al., 2021) and evaluate our method using three metrics: (1) Fidelity Distance (FD), which measures how well the input is preserved; (2) Minimal Matching Distance (MMD), which

Table 6 Quantitative comparison results on the PCN dataset in terms of per-point L1 Chamfer distance $\times 10^3$

Method	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft	Avg
PointNet++ (Qi et al., 2017b)	10.30	14.74	12.19	15.78	17.62	16.18	11.68	13.52	14.00
FoldingNet (Yang et al., 2018)	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31
AtlasNet (Groueix et al., 2018)	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61	10.85
PCN (Yuan et al., 2018)	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64
ForkNet (Wang et al., 2019)	9.08	14.22	11.65	12.18	17.24	14.22	11.51	12.66	12.85
TopNet (Tchapmi et al., 2019)	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15
MSN (Liu et al., 2020)	5.60	11.96	10.78	10.62	10.71	11.90	8.70	9.49	9.97
SoftPoolNet (Wang et al., 2020c)	6.93	10.91	9.78	9.56	8.59	11.22	8.51	8.14	9.20
GRNet (Xie et al., 2020b)	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
NSFA (Zhang et al., 2020a)	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48	8.06
CRN (Wang et al., 2020b)	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05	8.51
SCRN (Wang et al., 2021)	4.80	9.94	9.31	8.78	8.66	9.74	7.20	7.91	8.29
PMP-Net (Wen et al., 2021b)	5.50	11.10	9.62	9.47	6.89	10.74	8.77	7.19	8.66
PoinTr (Yu et al., 2021)	**4.05**	9.34	**7.97**	7.92	6.40	9.29	6.66	6.47	7.26
SnowflakeNet (Xiang et al., 2021)	4.29	**9.16**	8.08	7.89	6.07	**9.23**	**6.55**	6.40	7.21
SoftPool++ (Wang et al., 2022b)	5.50	10.02	8.73	9.05	7.53	10.24	8.01	7.43	8.31
Local Displacement (Wang et al., 2022a)	4.43	10.03	8.28	8.96	7.29	10.55	7.31	6.85	7.96
VQ-DcTr (Fei et al., 2022b)	4.51	10.15	8.82	8.34	6.23	10.46	7.14	6.38	7.75
PMP-Net++ (Wen et al., 2022)	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56
GTNet	4.17	9.33	8.38	**7.66**	**5.49**	9.44	6.69	**6.07**	**7.15**

The best values are highlighted in bold

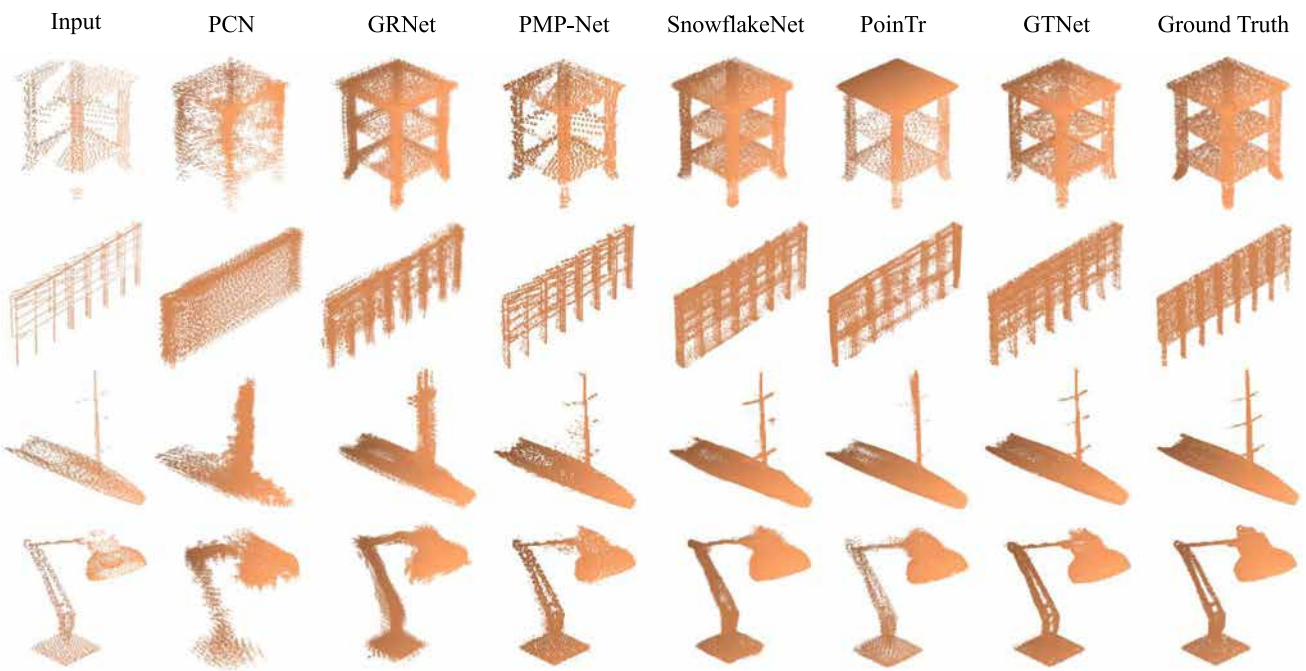

Fig. 8 Qualitative comparison results on the PCN dataset

Table 7 Quantitative comparison results on the KITTI dataset in terms of Fidelity Distance (FD) $\times 10^3$, Minimal Matching Distance (MMD) $\times 10^3$, and Consistency $\times 10^3$

Method	FD	MMD	Consistency
AtlasNet (Groueix et al., 2018)	1.759	2.108	0.700
FoldingNet (Yang et al., 2018)	7.467	0.537	1.053
PCN (Yuan et al., 2018)	2.235	1.366	1.557
TopNet (Tchapmi et al., 2019)	5.354	0.636	0.568
MSN (Liu et al., 2020)	0.434	2.259	1.951
GRNet (Xie et al., 2020b)	0.816	0.568	0.313
NSFA (Zhang et al., 2020a)	1.281	0.891	–
PFNet (Huang et al., 2020)	1.137	0.792	–
CRN (Wang et al., 2020b)	1.023	0.872	–
SpareNet (Xie et al., 2021)	1.461	0.368	0.249
PoinTr (Yu et al., 2021)	–	0.526	0.489
SnowflakeNet (Xiang et al., 2021)	0.034	0.407	–
SeedFormer (Zhou et al., 2022)	0.151	0.516	–
GTNet	**0.018**	**0.353**	**0.231**

The best results are highlighted in bold

measures how much the output resembles a typical car; (3) Consistency, which measures how consistent the network's outputs are against variations in the inputs. The quantitative comparison results with other state-of-the-art methods are shown in Table 7. Note that PoinTr (Yu et al., 2021) aims to predict the missing parts and concatenate the input point clouds as the final outputs, which results in the value of FD being zero. Therefore, we do not show FD for PoinTr in Table 7. Experimental results show that the proposed GTNet achieves excellent FD, MMD, and Consistency, which indicates that our method can complete the real point clouds with high stability and quality.

4.7 Ablation Studies

In this section, we first analyze the effectiveness of three key components of GTNet by adding or replacing a specific component, including geometric patch network, structure transformation network, and detail refinement network. The quantitative and qualitative comparison results are shown in Table 8 and Fig. 9, respectively. The baseline model A is PCN (Yuan et al., 2018) and the number of feature dimensions for global features is set to 512. Then, we analyze the effect of the translation matrix, different encoders, different metrics, and the number of points in the repetitive geometric structures on the performance of point cloud completion. Finally, we provide the robustness and detailed complexity analysis of our method. For convenience, we conduct ablation studies on the MVP dataset.

Table 8 Ablation studies on three key components of the proposed method including Feature Extraction (FE), Structure Transformation Network (STN), and Detail Refinement Network (DRN) on the MVP dataset ($1\times$ and $8\times$) in terms of per-point L2 Chamfer distance $\times 10^4$ and F-Score@1%

Model	FE	STN	DRN	2048($1\times$)		16,384($8\times$)	
				CD	F1	CD	F1
A				10.17	0.304	6.14	0.621
B	✓			9.01	0.375	5.29	0.673
C		✓		6.47	0.479	2.73	0.798
D			✓	8.66	0.399	5.16	0.687
E	✓	✓		6.19	0.487	2.42	0.811
F		✓	✓	6.04	0.496	2.33	0.815
G	✓		✓	8.21	0.422	5.02	0.704
H	✓	✓	✓	**5.76**	**0.510**	**2.19**	**0.823**

We report the mean CD and F-Score@1% for all categories. The best results are highlighted in bold

4.7.1 Effect of Geometric Patch Network

The Geometric Patch Network (GPN) is used to discover repetitive geometric structures based on the extracted local–global features. Effective local–global features of partial point clouds are the key to efficiently discover repetitive geometric structures. Feature Extraction (FE) module in GPN is used to obtain more effective local–global features. We analyze its effectiveness by using it to replace the PointNet-based feature extraction module in the baseline model. As denoted by Model A and Model B in Table 8, replacing the PointNet-based feature extraction module with FE reduces the average chamfer distance values by 11.4% and 13.8% at $1\times$ and $8\times$ resolutions, respectively. We further visualize the repetitive geometric structures discovered by GPN in Fig. 10. As can

Fig. 9 Visual comparison results of three key components in GTNet, including feature extraction module in geometric patch network, structure transformation network, and detail refinement network. Model A stands for the baseline model. GT stands for the ground truth of the 3D object

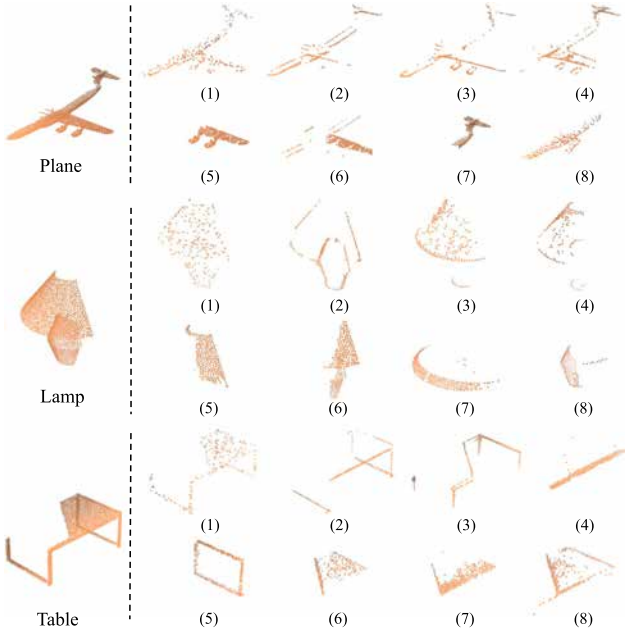

Fig. 10 Visualization of the repetitive geometric structures discovered by the score models in geometric patch network. Noted that here we only show the results of eight score models from a total of 32 score models at the resolution of 16384 points for obtaining better visualization presentation

be seen from the figure, the proposed GPN can effectively discover repetitive geometric structures of different scales in the partial point clouds. For example, for the plane shown in Fig. 10, the first four score models select different skeletons of it as repetitive geometric structures, while the remaining four score models select small-scale local structures such as wings, fuselage, and tail. All of these repetitive geometric structures can be used to complete the corresponding missing parts through symmetry.

4.7.2 Effect of Structure Transformation Network

The Structure Transformation Network (STN) learns a spatial transformation matrix for each repetitive geometric structure, which recovers the missing parts from patch-to-patch in a geometric transformation manner. To evaluate its effectiveness, we use it to replace the coarse generation module in the PCN decoder. As denoted by Model C in Table 8, STN improves the average chamfer distance values by 36.4% and 55.5% at $1\times$ and $8\times$ resolutions, respectively, demonstrating the effectiveness of completing missing parts in a geometric transformation manner. As denoted by Model E, F, G, and H in Table 8, we further evaluate the effectiveness of the combinations of the three components. The results show that the performance can be further improved when two or three components are used. In particular, we observe that the STN module contributes the most to the performance of our method. In Fig. 11, we also visualize the missing parts completion process of STN. We can find that STN selects different regions as repetitive geometric structures and then complete different missing parts through different spatial transformations.

4.7.3 Effect of Detail Refinement Network

When there are no repetitive geometric structures corresponding to the missing shapes in partial point clouds, the Detail Refinement Network (DRN) is used to generate the missing parts from the extracted local–global features. It is also used for global optimization to eliminate the unnatural structures caused by the overlaps among geometric patches. To evaluate its effectiveness, we use it to replace the refinement module in the PCN decoder. As denoted by Model D in Table 8, DRN improves the average chamfer distance values by 14.8% and 16.0% at $1\times$ and $8\times$ resolutions, which proves its effectiveness. As shown in Fig. 9, DRN can effectively eliminate unnatural structures, such as unevenly distributed

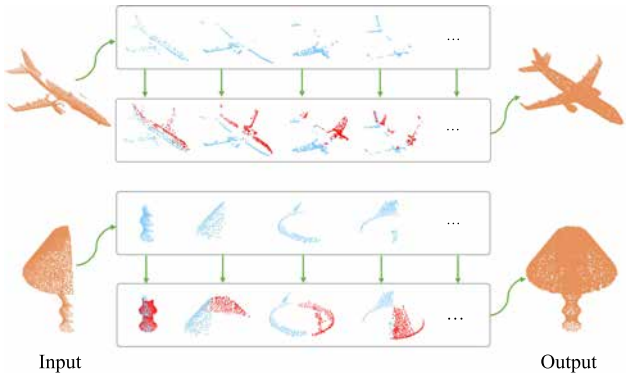

Fig. 11 Visualization of the repetitive geometric structures (highlighted in blue) discovered from partial point clouds and the corresponding missing shapes (highlighted in red) recovered by spatial transformations (Color figure online)

Table 9 Comparison of translation matrix $\mathbf{D}^m$ and translation vector $\mathbf{b}^m$ on the MVP dataset at various resolutions in terms of per-point L2 Chamfer distance $\times 10^4$

Points	2048 (1×)	4096 (2×)	8192 (4×)	16,384 (8×)
$\mathbf{b}^m$	6.20	4.59	3.34	2.41
$\mathbf{D}^m$	**5.76**	**4.18**	**3.05**	**2.19**

We report the mean CD for all categories. The best results are highlighted in bold

points and noise. Overall, the results show that the DRN is crucial for achieving high completion performance, as removing it leads to a significant degradation in results.

4.7.4 Effect of Translation Matrix

Since the translation vector $\mathbf{b}^m$ is the same for all points, it is impossible to correct the coordinate offset with different directions and magnitudes between the coarse output and the corresponding ground truth. As mentioned in Sect. 3.3, we propose to replace $\mathbf{b}^m$ with a translation matrix $\mathbf{D}^m$ to provide extra freedom of movement for each point. Table 9 shows the comparison results of the translation matrix $\mathbf{D}^m$ and the translation vector $\mathbf{b}^m$, which demonstrates that $\mathbf{D}^m$ significantly improves the completion performance at various resolutions. In fact, $\mathbf{b}^m$ can be regarded as a special case of $\mathbf{D}^m$. In this sense, $\mathbf{D}^m$ can naturally achieve better geometric transformations, which ultimately facilitates STN to efficiently complete the missing parts.

4.7.5 Effect of Different Encoders

As mentioned in Sect. 3.2.1, to design a simple network, we use a local encoder implemented by two shared MLPs with a max-pooling operation. To provide a detailed analysis of the effectiveness of different encoders, we replace the encoder of the proposed GTNet with three other encoders used in PoinTr (Yu et al., 2021), SoftPool++ (Wang et al., 2022b), and Local Displacement (Wang et al., 2022a). The quantitative comparison results on the MVP dataset (16384 points) are shown in Table 11, which shows that when the encoder of our GTNet is replaced with other encoders, the completion performance of our method can be further improved but at the cost of more complicated architectures (about two times of parameters than ours). In this paper, we use a simple encoder to achieve a better trade-off between model complexity and completion performance.

Table 10 Comparison of different K values on the MVP dataset (8×) in terms of per-point L2 Chamfer distance $\times 10^4$

K	$\frac{1}{2}N_p$	$\frac{1}{4}N_p$	$\frac{1}{8}N_p$	$\frac{1}{16}N_p$
CD	2.74	2.41	**2.19**	2.35
Time (ms)	37	76	151	299

We report the mean CD for all categories. The best results are highlighted in bold

4.7.6 Analysis of the Number of Points in the Repetitive Geometric Structure

To discover repetitive geometric structures, the Patch Generator (PG) module in GPN iteratively selects points with top-K scores in the partial point cloud as a repetitive geometric structure. As mentioned in Sect. 3.2.2, we set the number of points in each repetitive geometric structure to K for simplicity and efficiency. For this experiment, we analyze the effect of different K values on the completion performance. Experimental results are summarized in Table 10, which shows that $K = \frac{1}{8}N_p$ achieves the best performance. We can also note that as the number of points decreases, the time consumption increases significantly due to the increase in the number of iterations. However, the performance increases first and then decreases. The reason is that too many points result in repetitive geometric structures containing a lot of noise, while too few points can not form effective geometric structures. As shown in Fig. 12, an inappropriate K value makes PG unable to obtain effective repetitive geometric structures. By comparing different K values, we can find that an effective K value is critical to achieving better completion performance.

4.7.7 Analysis of Different Metrics

Chamfer Distance (CD), Earth Mover's Distance (EMD), and Density-aware Chamfer Distance (DCD) are broadly adopted metrics for measuring the similarity between two point sets. DCD is derived from the original CD, while it benefits from a higher sensitivity to distribution quality through a fraction term of query frequency and a higher tolerance to outliers through an approximation of Taylor Expansion. Following

Table 11 Quantitative comparison results of different encoders on the MVP dataset (16,384 points) in terms of per-point L2 Chamfer distance $\times 10^4$ and Params (M)

Method	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Avg	Params
Our GTNet	0.64	**3.08**	2.15	2.73	2.46	2.90	2.72	1.85	2.19	**11.2**
-with PoinTr (Yu et al., 2021)	0.65	3.20	2.16	**2.67**	2.41	**2.87**	**2.68**	1.87	2.16	21.3
-with SoftPool++ (Wang et al., 2022b)	0.62	3.13	**2.13**	2.70	2.43	2.91	2.69	1.84	2.04	19.6
-with Local Displacement (Wang et al., 2022a)	**0.61**	3.09	2.14	2.68	**2.40**	2.89	2.71	**1.83**	**1.91**	23.7

We report the detailed results of 8 categories and the average results of 16 categories. The best results are highlighted in bold

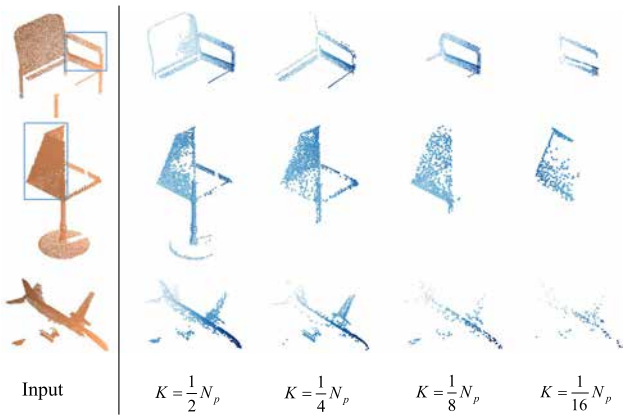

Fig. 12 Illustration of the number of points in the repetitive geometric structure. A reasonable K value is critical to obtain effective repetitive geometric structures

previous work (Wu et al., 2021), we evaluate the effectiveness of different metrics by training networks with each of the three metrics (denoted by L_{CD}, L_{EMD}, and L_{DCD}) and evaluate on all of them (denoted by CD, EMD, and DCD), as shown in Table 12. Compared to networks trained with L_{CD}, training with L_{EMD} achieves the lowest EMD and lower DCD, but it has the highest CD. We also find that training with L_{DCD} achieves the lowest DCD, significantly reduces EMD, and can even slightly reduce CD than the L_{DCD}-trained networks.

4.7.8 Robustness to Occlusion

To evaluate the robustness of the proposed GTNet, we conduct experiments on the input point clouds with various occlusion ratios. Following PCN (Yuan et al., 2018) and CRN (Wang et al., 2020b), we manually occlude the partial inputs from the testing dataset by p percent of points, where p ranges from 20% to 80% with a step of 10%. The quantitative and qualitative comparison results are shown in Table 13 and Fig. 13, respectively. GTNet achieves the best completion performance at all occlusion rates, which demonstrates that our method has better robustness in handling incomplete inputs with varying degrees of missingness.

Table 12 Quantitative comparison for three metrics when trained with each of them on the MVP dataset (16,384 points) in terms of CD $\times 10^4$, EMD $\times 10^2$, and DCD

Metric	Loss Function		
	L_{CD}	L_{EMD}	L_{DCD}
CD	2.19	3.47	**2.15**
EMD (Liu et al., 2020)	5.88	**3.25**	4.63
DCD (Wu et al., 2021)	0.51	0.48	**0.42**

The best results are highlighted in bold

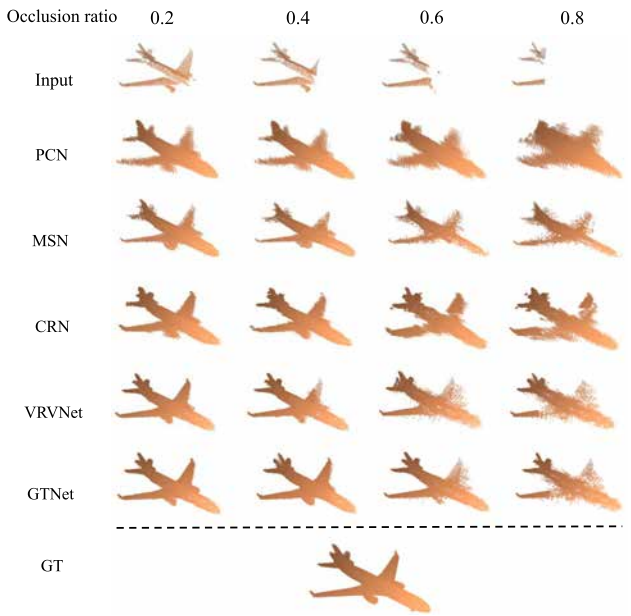

Fig. 13 Qualitative comparison on inputs with different occlusion ratios. The proposed GTNet exhibits better robustness in handling incomplete inputs with varying degrees of missingness

4.7.9 Complexity Analysis

The proposed GTNet achieves the best performance on both synthetic benchmarks and real-scanned benchmarks. In this subsection, we further provide a detailed complexity analysis of the proposed method. As shown in Table 14, we report the number of parameters and FLOPs of our method and several other state-of-the-art methods. From the table, we can see that the proposed GTNet has the least parameters while using

Table 13 Quantitative comparison for occluded point clouds under different occlusion rates on the MVP dataset (16,384 points) in terms of per-point L2 Chamfer distance $\times 10^4$

Method	Occlusion ratios						
	%20	%30	%40	%50	%60	%70	%80
PCN (Yuan et al., 2018)	6.35	6.61	7.04	7.45	8.07	8.52	9.55
MSN (Liu et al., 2020)	5.54	5.77	6.21	6.59	7.14	7.65	8.79
CRN (Wang et al., 2020b)	4.78	5.11	5.43	5.78	6.31	6.93	8.17
VRCNet (Pan et al., 2021)	3.38	3.73	4.11	4.43	4.79	5.44	7.04
GTNet	**2.37**	**2.46**	**2.79**	**3.25**	**3.77**	**4.52**	**6.33**

We report the mean CD for all categories. The best results are highlighted in bold

Table 14 The number of parameters (Params) and FLOPs of different methods on the PCN dataset

Method	Params (M)	FLOPs (G)
GRNet (Xie et al., 2020b)	76.7	26.9
SpareNet (Xie et al., 2021)	82.6	73.2
SnowflakeNet (Xiang et al., 2021)	19.2	**9.9**
PoinTr (Yu et al., 2021)	42.5	11.2
VRCNet (Pan et al., 2021)	17.5	29.8
SoftPool++ (Wang et al., 2022b)	61.7	24.7
GTNet	**11.2**	13.7

The best results are highlighted in bold

relatively low FLOPs, which further proves the simplicity and effectiveness of GTNet.

5 Conclusion

In this paper, we formulate point cloud completion as a geometric transformation problem and propose a novel Geometric Transformation Network (GTNet) to complete the missing parts. The main motivation of this work is to better exploit repetitive geometric structures in common 3D objects to complete point clouds by transforming geometric structures from the observed parts to the unobserved parts in a spatial transformation manner. To this aim, we propose the geometric patch network to discover repetitive geometric structures and the structure transformation network to transform repetitive geometric structures. We also propose the detail refinement network for global optimization, which eliminates unnatural structures. Extensive comparisons are conducted on the ShapeNet55-34, MVP, PCN, and KITTI datasets, which indicate that the proposed GTNet achieves superior results compared to the state-of-the-art methods.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Nos. 62272134 and 62236003), in part by the Taishan Scholars Program of Shandong Province (No. tsqn201812106), in part by the Shenzhen Colleges and Universities Stable Support Program (No. GXWD20220817144428005), in part by the National Key R&D Program of China (No. 2021ZD0110901).

Data Availability All datasets generated or analyzed during the current study are included in the published articles Yu et al. (2021); Pan et al. (2021). These datasets can be derived from the following public-domain resources: https://github.com/paul007pl/VRCNet, https://github.com/yuxumin/PoinTr/blob/master/DATASET.md.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., & Guibas, L. (2018). Learning representations and generative models for 3D point clouds. In *International conference on machine learning* (pp. 40–49).

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223).

Buades, A., Duran, J., & Navarro, J. (2019). Motion-compensated spatio-temporal filtering for multi-image and multimodal super-resolution. *International Journal of Computer Vision, 127*(10), 1474–1500.

Chang, A. X., Funkhouser, T., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., & Yu, F. (2015). Shapenet: An information-rich 3D model repository. arXiv:1512.03012

Chen, X., Chen, B., & Mitra, N. J. (2020). Unpaired point cloud completion on real scans using adversarial training. In *International conference on learning representations*.

Chrysos, G. G., Kossaifi, J., & Zafeiriou, S. (2020). RoCGAN: Robust conditional GAN. *International Journal of Computer Vision, 128*(10), 2665–2683.

Dai, A., Ruizhongtai Qi, C., Nießner, M. (2017). Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5868–5877).

Fei, B., Yang, W., Chen, W. M., & Ma, L. (2022b). VQ-DcTr: Vector-quantized autoencoder with dual-channel transformer points splitting for 3D point cloud completion. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 4769–4778).

Fei, B., Yang, W., Chen, W., Li, Z., Li, Y., Ma, T., Hu, X., & Ma, L. (2022a). Comprehensive review of deep learning-based 3D point clouds completion processing and analysis. arXiv preprint arXiv:2203.03311

Fu, Y., Lam, A., Sato, I., & Sato, Y. (2017). Adaptive spatial-spectral dictionary learning for hyperspectral image restoration. *International Journal of Computer Vision, 122*(2), 228–245.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research, 32*(11), 1231–1237.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Graham, B., Engelcke, M., & van der Maaten, L. (2018). 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9224–9232).

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 216–224).

Guo, M. H., Cai, J. X., Liu, Z. N., Mu, T. J., Martin, R. R., & Hu, S. M. (2021). Pct: Point cloud transformer. *Computational Visual Media*, *7*(2), 187–199.

Han, X., Li, Z., Huang, H., Kalogerakis, E., & Yu, Y. (2017) 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* (pp. 85–93).

Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X. (2020). PF-Net: Point fractal network for 3D point cloud completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7662–7670).

Hui, L., Xu, R., Xie, J., Qian, J., & Yang, J. (2020). Progressive point cloud deconvolution generation network. In *European conference on computer vision* (pp. 397–413).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*.

Klokov, R., & Lempitsky, V. (2017). Escape from cells: Deep KD-networks for the recognition of 3D point cloud models. In Proceedings of the IEEE international conference on computer vision (pp. 863–872).

Li, R., Li, X., Fu, C. W., Cohen-Or, D., & Heng, P. A. (2019). PU-GAN: A point cloud upsampling adversarial network. In *Proceedings of the IEEE international conference on computer vision* (pp. 7203–7212).

Liu, M., Sheng, L., Yang, S., Shao, J., & Hu, S. M. (2020) Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11,596–11,603).

Lyu, Z., Kong, Z., Xu, X., Pan, L., & Lin, D. (2022). A conditional point diffusion-refinement paradigm for 3D point cloud completion. In *International conference on learning representations*.

Mark, P., Niloy, J. M., Johannes, W., Helmut, P., & Leonidas, J. G. (2008). Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, *27*(3), 43.

Niloy, J. M., Leonidas, J. G., & Mark, P. (2006). Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics*, *25*(3), 560–568.

Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., & Liu, Z. (2021). Variational relational point completion network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8524–8533).

Pan, L. (2020). ECG: Edge-aware point cloud completion with graph convolution. *IEEE Robotics and Automation Letters*, *5*(3), 4392–4398.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019) Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (pp. 8024–8035).

Qi, C. R., Su, H., Mo, K., Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems* (pp. 5099–5108).

Riegler, G., Osman Ulusoy, A., & Geiger, A. (2017). OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3577–3586).

Sarmad, M., Lee, H. J., Kim, Y. M. (2019). RL-GAN-Net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5898–5907).

Schiebener, D., Schmidt, A., Vahrenkamp, N., & Asfour, T. (2016). Heuristic 3D object shape completion based on symmetry and scene context. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 74–81).

Shu, D. W., Park, S. W., & Kwon, J. (2019) 3D point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3859–3868).

Sipiran, I., Mendoza, A., Apaza, A., & Lopez, C. (2022). Data-driven restoration of digital archaeological pottery with point cloud analysis. *International Journal of Computer Vision*, *130*, 1–17.

Stutz, D., & Geiger, A. (2018). Learning 3D shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1955–1964).

Stutz, D., & Geiger, A. (2020). Learning 3D shape completion under weak supervision. *International Journal of Computer Vision*, *128*(5), 1162–1181.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M. H., & Kautz, J. (2018). Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2530–2539).

Sung, M., Kim, V. G., Angst, R., & Guibas, L. (2015). Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, *34*(6), 1–11.

Tatarchenko, M., Dosovitskiy, A., & Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proceedings of the IEEE international conference on computer vision* (pp. 2088–2096).

Tchapmi, L. P., Kosaraju, V., Rezatofighi, H., Reid, I., & Savarese, S. (2019). Topnet: Structural point cloud decoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 383–392).

Thrun, S., & Wegbreit, B. (2005). Shape from symmetry. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1824–1831).

Wang, X., Ang, Jr. M. H, & Lee, G. H. (2020b). Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 790–799).

Wang, X., Ang, Jr. M. H., & Lee, G. H. (2020a). Point cloud completion by learning shape priors. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 10,719–10,726).

Wang, Y., Tan, D. J., Navab, N., & Tombari, F. (2019). Forknet: Multi-branch volumetric semantic completion from a single depth image. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8608–8617).

Wang, Y., Tan, D. J., Navab, N., & Tombari, F. (2020c). Softpoolnet: Shape descriptor for point cloud completion and classification. In *European conference on computer vision* (pp. 70–85). Springer.

Wang, Y., Tan, D. J., Navab, N., & Tombari, F. (2022a). Learning local displacements for point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1568–1577).

Wang, X., Ang, M. H., & Lee, G. H. (2021). Cascaded refinement network for point cloud completion with self-supervision.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11), 8139–8150.

Wang, P. S., Sun, C. Y., Liu, Y., & Tong, X. (2018). Adaptive ocnn: A patch-based deep representation of 3d shapes. *ACM Transactions on Graphics, 37*(6), 1–11.

Wang, Y., Tan, D. J., Navab, N., & Tombari, F. (2022). Softpool++: An encoder-decoder network for point cloud completion. *International Journal of Computer Vision, 130*(5), 1145–1164.

Wen, X., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S, (2021a). Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13,080–13,089).

Wen, X., Li, T., Han, Z., & Liu, Y. S. (2020). Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1939–1948).

Wen, X., Xiang, P., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S. (2021b). Pmp-net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7443–7452).

Wen, X., Xiang, P., Han, Z., Cao, Y. P., Wan, P., Zheng, W., & Liu, Y. S. (2022). Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wu, R., Chen, X., Zhuang, Y., & Chen, B. (2020). Multimodal shape completion via conditional generative adversarial networks. In *European conference on computer vision* (pp. 281–296).

Wu, T., Pan, L., Zhang, J., Wang, T., Liu, Z., & Lin, D. (2021). Density-aware chamfer distance as a comprehensive metric for point cloud completion. In *Advances in neural information processing systems*.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1912–1920).

Xia, Y., Xia, Y., Li, W., Song, R., Cao, K., & Stilla, U. (2021). Asfm-net: Asymmetrical siamese feature matching network for point completion. In *Proceedings of the 29th ACM international conference on multimedia* (pp. 1938–1947).

Xiang, P., Wen, X., Liu, Y. S., Cao, Y. P., Wan, P., Zheng, W., & Han, Z. (2021). Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE international conference on computer vision* (pp. 5499–5509).

Xie, C., Wang, C., Zhang, B., Yang, H., Chen, D., & Wen, F. (2021). Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (pp. 4619–4628).

Xie, H., Yao, H., Sun, X., Zhou, S., & Zhang, S. (2019). Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2690–2698).

Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., & Sun, W. (2020b) GRNet: Gridding residual network for dense point cloud completion. In *European conference on computer vision* (pp. 365–381).

Xie, H., Yao, H., Zhang, S., Zhou, S., & Sun, W. (2020). Pix2Vox++: Multi-scale context-aware 3D object reconstruction from single and multiple images. *International Journal of Computer Vision, 128*(12), 2919–2935.

Yan, W., Zhang, R., Wang, J., Liu, S., Li, T. H., & Li, G. (2020). Vaccine-style-net: Point cloud completion in implicit continuous function space. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 2067–2075).

Yang Y., Feng C., Shen Y., & Tian, D. (2018) Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 206–215).

Yang, Y., Cao, Q., Zhang, J., & Tao, D. (2022). Codon: On orchestrating cross-domain attentions for depth super-resolution. *International Journal of Computer Vision, 130*(2), 267–284.

Yu X., Rao Y., Wang Z., Liu Z., Lu J., & Zhou, J. (2021). PoinTr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE international conference on computer vision* (pp. 12,498–12,507).

Yuan, W., Khot, T., Held, D., Mertz, C., & Hebert, M. (2018). PCN: Point completion network. In *International conference on 3D vision (3DV)* (pp. 728–737).

Zhang, W., Yan, Q., & Xiao, C. (2020a). Detail preserved point cloud completion via separated feature aggregation. In *European conference on computer vision* (pp. 512–528).

Zhang, X., Dong, H., Hu, Z., Lai, W. S., Wang, F., & Yang, M. H. (2020). Gated fusion network for degraded image super resolution. *International Journal of Computer Vision, 128*(6), 1699–1721.

Zhang, H., Li, Y., Chen, H., Gong, C., Bai, Z., & Shen, C. (2022). Memory-efficient hierarchical neural architecture search for image restoration. *International Journal of Computer Vision, 130*(1), 157–178.

Zhao, Y., Zhou, Y., Chen, R., Hu, B., & Ai, X. (2021). MM-Flow: Multimodal flow network for point cloud completion. In *Proceedings of the 29th ACM international conference on multimedia* (pp. 3266–3274).

Zheng, C., Cham, T. J., & Cai, J. (2021). Pluralistic free-form image completion. *International Journal of Computer Vision, 129*(10), 2786–2805.

Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., & Wang, C. (2022). Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision* (pp. 416–432).

Zhou, L., Du, Y., & Wu, J. (2021). 3D shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5826–5835).