Dense Point Cloud Completion Based on Generative Adversarial Network

Ming Cheng, *Member, IEEE*, Guoyan Li, Yiping Chen, *Senior Member, IEEE*, Jun Chen, Cheng Wang, *Senior Member, IEEE*, and Jonathan Li, *Senior Member, IEEE*

***Abstract*— Point cloud completion aims to reconstruct complete point clouds from partial point clouds, which is widely used in various fields such as autonomous driving and robotics. Most existing methods are sparse point cloud completion, where the number of point clouds after completion is relatively small and the details are insufficient. This article proposes a novel end-to-end generative adversarial network-based dense point cloud completion architecture (DPCG-Net). We design two generative adversarial network (GAN)-based modules that translate point cloud completion into mapping between global feature distributions obtained by encoding partial point clouds and ground truth, respectively. The first designed generator module proposes skip connections to fully connected layer-based network for regenerating global feature and changing the global feature distribution derived from the encoder module to approximate the ground truth global feature distribution. The second proposed discriminator module divides high-dimensional global feature vectors into several smaller batches for judgment to guarantee the similarity between the regenerated global feature and the ground truth. We perform quantitative and qualitative experiments on the ShapeNet and KITTI datasets. Experiments on ShapeNet demonstrate that our model outperforms other models in cases where the lack of a large proportion of point clouds results in a large loss of spatial structure, especially when 80% of point clouds are missing. Moreover, KITTI experiments reveal that it is also valid for realistic situations. In addition, application in classification shows that the classification accuracy of point clouds completed with DPCG-Net is as high as 86.5% under the condition of 80% missing point clouds.**

***Index Terms*— 3-D point cloud, deep learning, generative adversarial network (GAN), shape completion.**

I. Introduction

POINT clouds are the most commonly used 3-D data format, which can maintain the original geometric information of objects in 3-D space and are widely used in many fields such as digital preservation, architecture, 3-D games, robotics, and virtual reality [1]. Point clouds can be acquired by using laser scanners, stereo cameras, or low-cost RGB-D scanners. In the process of point cloud collection, due to factors such as occlusion of sensor resolution and viewing angle limitations, geometric and semantic information of object will be lost, resulting in incomplete data obtained by scanning [2]. Point cloud completion is necessary for better application of point clouds to subsequent fields, such as robotics and digital industries.

In recent years, deep learning has made great progress in many areas [3]–[5], while more and more large 3-D CAD models, such as ShapeNet [6] dataset, have been released, providing powerful technology and data to support point cloud completion. Point clouds have the nature of disorder and are difficult to be directly used in convolutional neural networks. Many researchers voxelize point clouds and use convolutional neural networks for training [7]–[10]. However, as the resolution of 3-D voxel grids increases, computational memory consumption is huge and it is difficult to handle detailed information.

With the advent of PointNet and its extended networks [11], [12], it is possible to train directly on point clouds. The mainstream research methods are sparse point cloud completion such as SA-Net [13], RL-GAN-Net [14], and PF-Net [2], which means that the number of point clouds after completion does not exceed 2048. Dense point clouds contain more detailed features than sparse point clouds, but the increase in the number of point clouds requires more computational cost, and it is a big challenge to handle and generate denser point clouds, especially when a large percentage of point clouds are missing. There is a relatively short history of dense point cloud completion. To the authors' best knowledge, PCN [15] first proposes a two-stage approach to achieve dense point cloud completion, and the number of point clouds is 16 384 after completion. TopNet [16] proposes a hierarchical tree-structured decoder to implement completion. However, these methods pay less attention to completion in the case of larger proportional structural loss of point clouds.

To address the challenge, we propose a new approach to dense point cloud completion using generative adversarial network (GAN) inspired by Isola *et al.*'s work [17] in the image domain. GAN aims to create a style transfer between the high-dimensional global feature vector of partial point clouds and that of ground truth. Generator adopts a fully connected layer-based network combined with skip connections [18],

Manuscript received June 7, 2021; revised July 19, 2021; accepted August 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U1605254 and Grant 41871380, and in part by the Natural Sciences and Engineering Research Council of Canada under Grant 50503-10284. *(Corresponding author: Yiping Chen.)*

Ming Cheng, Guoyan Li, Yiping Chen, Jun Chen, and Cheng Wang are with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: chm99@xmu.edu.cn; liguoyan@stu.xmu.edu.cn; chenyiping@xmu.edu.cn; chenjun97@stu.xmu.edu.cn; cwang@xmu.edu.cn).

Jonathan Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Digital Object Identifier 10.1109/TGRS.2021.3105551

which is responsible for transforming the high-dimensional global feature vector of partial point clouds into a new feature vector. Discriminator is a fully connected layer-based network inspired by PatchGAN [17], which is used to distinguish whether the input vector is from generator or ground truth. GAN-based dense point cloud completion architecture (DPCG-Net) transforms the problem of mapping between partial point clouds and ground truth into a mapping between their high-dimensional global feature vectors, in which the proposed generator continuously normalizes the process by learning in two different feature vector spaces. Regenerated feature vectors obtained by the generator can be used by decoder to achieve dense point cloud completion. Our main contributions are as follows.

1) We propose a novel end-to-end DPCG-Net for completing noisy and partial 3-D point clouds. This novel architecture is able to obtain the dense point clouds from incomplete point clouds effectively with even an 80% missing rate.
2) We design a novel generator module after the encoder step to regenerate the global feature of partial point clouds, which lies within the learned ground truth global feature distribution and is close to the partial point clouds global feature.
3) We design a new discriminator module to divide the high-dimensional global feature vector into several smaller batches for judgment, which guarantees the similarity between the regenerated partial point clouds global feature and the ground truth.

II. Related Work

A. Voxel-Based Method

Voxelized point clouds are convenient for shape completion using convolutional neural networks. Wu *et al.* [7] proposed 3-D ShapeNets, which used convolutional deep belief network (CDBN) to represent the 3-D geometric shapes as probability distributions of binary variables on a 3-D voxel grid. Using geometric data and multiview RGB data, Nguyen *et al.* [8] employed the Markov random field (MRF) model to complement point clouds using geometric and multiview RGB data. Han *et al.* [9] proposed a completion algorithm that combined overall structural information and local geometric information. Dai *et al.* [19] completed partial 3-D shapes by combining volumetric deep neural networks and 3-D shape synthesis. Xie *et al.* [20] proposed a gridding residual network (GRNet). The network regularized disordered point clouds to the 3-D grid, so as to better obtain the geometric structure and context information of point clouds. Nevertheless, these methods typically perform convolution in a regular voxel grid, which requires high memory and computational costs.

Researchers have begun to explore the representation of sparse voxels and structural design of irregular convolutions. Riegler *et al.* [21] exploited the sparsity of input data by using an unbalanced octree representation of data, allowing for deeper networks without compromising resolution. Wang *et al.* [10] presented an adaptive octree-based convolutional neural network (Adaptive O-CNN), which adaptively represented a 3-D shape with different levels of octants and modeled the 3-D shape with a planar patch within each octant. Based on their work [10], [21], Wang *et al.* [22] introduced a new output-guided skip connection in the network structure that can better retain the input geometric information and effectively learn geometric priors from the data. Graham *et al.* [23] proposed submanifold sparse convolutional networks (SSCNs) for sparse data. They defined novel sparse convolutional operations that can handle sparse data more efficiently and implement spatially sparse convolutional networks. Although there are many improved algorithms that have achieved surprising results, computational cost proportional to the resolution of input data still makes it difficult to handle the fine textures of data.

B. Point-Based Method

Point-based methods benefit from recent advances in deep neural networks that operate directly on point clouds, such as PointNet [11] and PointNet++ [12]. Many existing research methods focus on sparse point cloud completion, where the number of point clouds after completion does not exceed 2048. Yang *et al.* [24] proposed an end-to-end autoencoder network, which introduced a decoder based on the folding operation to deform canonical 2-D meshes to 3-D object surfaces. Wen *et al.* [13] used a skip-attention mechanism to send feature information to different stages of generation to realize point cloud completion. In addition, many researchers have introduced GAN into the study of sparse point cloud completion. Achlioptas *et al.* [25] trained GAN in the latent feature space and remapped these latent features to generate 3-D point clouds. Gurumurthy and Agrawal [26] proposed an initialized encoder to bridge autoencoder and GAN trained in the latent feature space to achieve point cloud completion. Sarmad *et al.* [14] applied reinforcement learning to construct a correspondence between the global feature vector of partial point clouds and the input random noise vector of GAN to complement point clouds. Huang *et al.* [2] proposed a multiscale hierarchical GAN called PF-Net, which introduced a discriminator to evaluate the quality of point clouds. Chen *et al.* [27] exploited unpaired data to train GAN for point cloud completion, which can be better applied to realistic scenarios. However, the point clouds are relatively sparse after completion.

Dense point clouds contain richer detail features, but there are relatively few existing studies on dense point cloud completion. To the best of our knowledge, Yuan *et al.* [15] first proposed a dense point cloud completion algorithm called PCN with the number of completed point clouds up to 16 384. This algorithm provided a two-stage completion approach to achieve point cloud completion. Tchapmi *et al.* [16] presented a point cloud completion method with the hierarchical tree structure for decoder, named Topnet. Subsequent methods also adopted the multistage strategy of coarse-to-fine similar to that in PCN. Peng *et al.* [28] proposed SDME-Net, which was used for uniform completion of unstructured point clouds. They added repulsion loss to the loss function to make the generated point clouds more uniformly distributed.

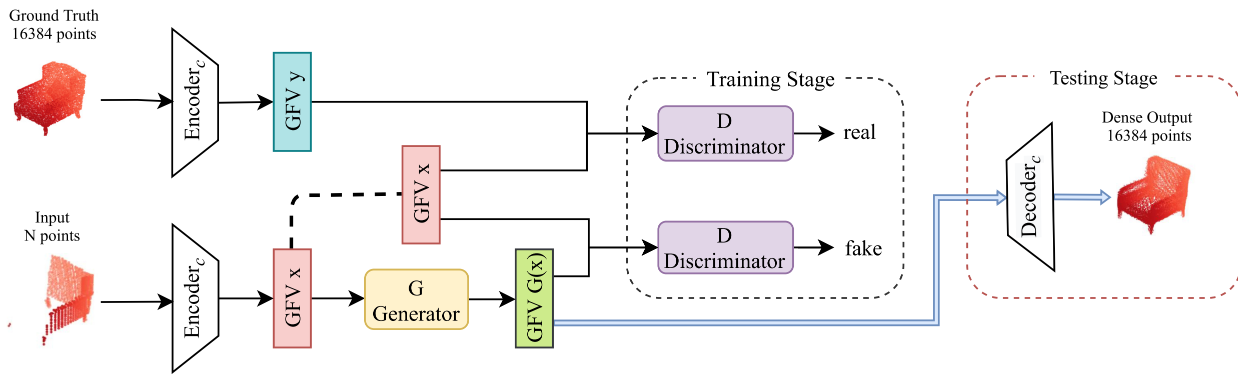


Fig. 1. Overall network structure of DPCG-Net algorithm. GAN is trained in the global feature vector space obtained from trained encoder (Encoder$_c$). As shown by double-line blue arrows in the figure, the vector [GFV $G(x)$] regenerated by generator is fed to trained decoder (Decoder$_c$) to obtain dense point clouds.

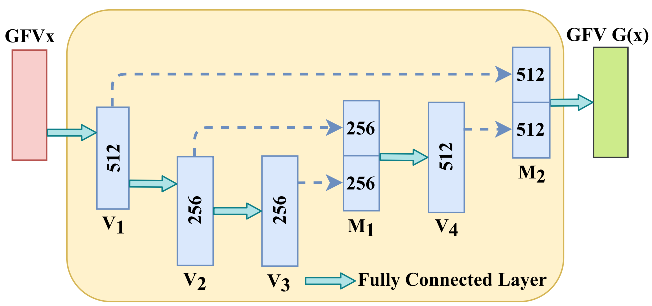


Fig. 2. Network structure of generator in DPCG-Net.

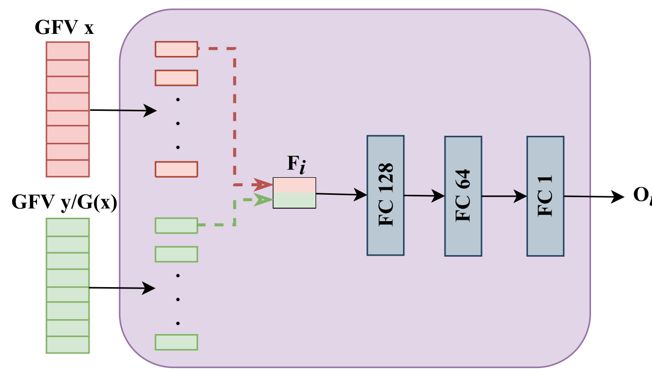


Fig. 3. Network structure of discriminator in DPCG-Net.

Zhang *et al.* [29] showed a multistage point completion network (MSPCN) with critical set supervision, which utilized critical sets for supervision and produced informative and useful intermediate outputs for the next stage. Liu *et al.* [30] obtained coarse-grained prediction with the input point clouds at first stage, which was subsequently combined with the original input to learn a point-wise residual for fine-grained details of point clouds. Wang *et al.* [31] proposed a cascading shape completion algorithm to synthesize local and global information of missing point clouds to generate high-quality point clouds. In addition, there are other approaches. Yan *et al.* [32] presented a method for completion in the function space of a 3-D surface, which embedded a reinforcement learning agent to generate the complete output. Zhang *et al.* [33] used multilevel feature extraction and separated feature aggregation to improve the problem of detail loss in the completion process. Son and Kim [34] adopted traditional methods to predict unknown parts and added a symmetry-aware upsampling module (SAUM) to exploit symmetries for shape completion. There are relatively few dense point clouds completion algorithms, and the completeness of dense point cloud completion is still worth investigating.

III. Method

The structure of our proposed DPCG-Net is shown in Fig. 1, which is inspired by the GAN-based network and consists of two main innovative modules. The designed generator module and discriminator module are shown in Figs. 2 and 3, respectively. First, we train an encoder–decoder model with partial point clouds and ground truth. The feature vector is relatively noisy when partial point clouds are fed into a trained encoder (Encoder$_c$) to obtain a high-dimensional global feature vector (GFV x), whereas the high-dimensional global feature vector (GFV y) of ground truth is cleaner. Then, we propose a GAN-based method including generator module and discriminator module that aims to establish a distribution transfer between the high-dimensional global feature vector of partial point clouds and ground truth. Regenerated vector [GFV $G(x)$] obtained by generator is supplied to trained decoder (Decoder$_c$) for generating dense point clouds.

A. Encoder–Decoder

Encoder–decoder model utilizes our previously proposed N-DPC [35] network architecture. Encoder is PointNet-based network incorporating self-attention mechanism [36], [37] for extracting global features of point clouds. In the encoder stage, the number of input point clouds varies for different samples. PointNet-based networks can deal with a different number of point clouds and extract features of point clouds. Given N points as input, shared multilayer perceptrons (MLPs) with a structure of [128, 256] generate feature f with a dimension of $(N \times 256)$. Each row in f represents the feature of a point, and the global feature vector g is obtained through the maxpool operation. f is connected to g to produce feature vector F of dimension ($N \times 512$). Then, F is fed to the self-attention attention mechanism to reassign the weights of points' features to obtain F', with the same dimension as F. Finally, global feature vector G is obtained by shared MLPs with structure [512, 1024] and maxpool operation, and the dimension of G is (1×1024). Compared to the global feature vector in other algorithms that combine GAN (e.g., RL-GAN-Net [14] with dimension 128), the global feature vector of our model has a dimension of 1024, which is relatively high and is called high-dimensional global feature vector.

A decoder contains coarse completion and dense completion, which is responsible for reconstructing the high-dimensional global feature vector to generate dense point clouds. The coarse completion module consists of three fully connected layers with output dimensions of [1024, 1024, $M \times 3$], where M denotes the number of coarse point clouds. The dense completion module first extracts the local features of coarse point clouds using a PointNet++ [12] structure, which consists of shared MLPs of [64, 128, 256] and neighborhood feature pooling operation. The obtained local feature vector L is of dimension ($M \times 256$). Then, L is concatenated with 3-D coordinate information C of coarse point clouds and the high-dimensional global feature vector G to get the joint feature, which is tiled 16 times to obtain the feature $\{L + C + G\}$. In order to distinguish duplicate points, we introduce the 2-D grid data to obtain feature $\{L + C + G + 2\}$, which is put into shared MLPs with structure [512, 512, 3] and merged with the coordinates C of coarse point clouds to finally generate dense point clouds with the number of 16 384. The loss function Loss_{E-D} is a combination of chamfer distance (CD) and Earth mover's distance (EMD) [38] and is described as follows:

$$\text{Loss}_{E-D} = \text{EMD}\left(P_{\text{coarse}}, P'_{gt}\right) + \alpha \text{CD}\left(P_{\text{dense}}, P_{gt}\right) \tag{1}$$

where EMD calculates the distance between coarse point clouds P_{coarse} and coarse ground truth P'_{gt}. CD measures the distance between dense point clouds P_{dense} and dense ground truth P_{gt}. α donates the constant coefficient of CD, which is not greater than 1 during training stage. We train this model on the training set. With the trained encoder, we can obtain the high-dimensional global feature vector of point clouds and then train GAN in that vector space.

B. Structure of GAN

1) Generator: This module uses the high-dimensional global feature vector (GFV x) obtained by the encoder from partial point clouds to regenerate a new high-dimensional global feature vector [GFV $G(x)$]. The specific structure of generator is shown in Fig. 2, where double-line arrows represent fully connected layers. The design of generator draws on the idea of skip connections of U-Net [18] to realize information sharing between input and output. Unlike U-Net, which uses convolutional neural nets in the image space, our generator applies fully connected layers in the feature vector space and uses skip connections between different layers to regenerate new vectors.

First, the high-dimensional global feature vector (GFV x) obtained from partial point clouds by the trained encoder is fed to three fully connected layers with output dimensions of [512, 256, 256] to generate feature vectors V_1, V_2, and V_3, respectively. Then, the feature vectors V_2 and V_3 are merged to obtain the feature vector M_1 of dimension 512. With M_1 as input, the fully connected layer generates feature vector V_4 of dimension 512. Also, V_1 concatenates V_4 to obtain intermediate vector M_2. Finally, the high-dimensional global feature vector [GFV $G(x)$] is obtained by M_2 through a fully connected layer with an output dimension of 1024, which is subsequently fed to the trained decoder for dense point cloud completion.

2) Discriminator: This module is designed to determine whether the high-dimensional global feature vector comes from ground truth or generator. The network structure of discriminator is shown in Fig. 3. Since the global feature vector of point clouds obtained by the encoder has a higher dimension (1024 dimensions), this article proposes PatchGAN's idea to design a discriminator, which is commonly used in the image domain. Compared with regular discriminators that directly determine real/fake of the whole image, PatchGAN maps the image to a probability matrix by a convolutional neural network, and each value in the matrix corresponds to real/fake of each $N \times N$ region in the original image.

We manually cut the high-dimensional feature vector into ($1 \times d$) regions and judge them with regular discriminator. Then, all results are averaged to evaluate the entire input vector. Specifically, we first divide the 1024-D global feature vector into each feature vector of dimension d and set ($d = 128$) during the experiment. In order to reduce the calculation cost, each small feature vector does not overlap and it is divided into $[1:128, 129:256, \ldots, 897:1024]$, a total of eight feature vectors ($128 \times 8 = 1024$). Then, the small feature vectors are concatenated separately to obtain $\{Fi | i = 1, 2, \ldots, 8\}$, which is judged by a regular discriminator based on fully connected layers to obtain $\{O_i | i = 1, 2, \ldots, 8\}$. Finally, all results are averaged to obtain the final output, where the structure of discriminator is fully connected layers with output dimensions of [128, 64, 1].

C. Loss Function of GAN

The training of generator G and discriminator D is based on the high-dimensional global feature vector space of point clouds. Let X represent partial point clouds, $x = \text{Encoder}_C(X)$ represent the high-dimensional global feature vector of partial point clouds, Y represent ground truth, and $y = \text{Decoder}_c(Y)$ represent the high-dimensional global feature of ground truth. The input of G is x, which also changes even when there is no noise, so noise is no longer added here [39]. First, we introduce the training of discriminator D. D classifies the one-to-one corresponding global feature vector pair (x, y) as 1 (real) and discriminates $[x, G(x)]$ as 0 (fake). At this stage, the weight of generator G is fixed, and the weight of D is updated iteratively. The loss function of discriminator Loss_D is as follows:

$$\text{Loss}_D = 0.5 L_{\text{bce}}\left(D(x, y), 1\right) + 0.5 L_{\text{bce}}\left(D(x, G(x)), 0\right) \tag{2}$$

$$L_{\text{bce}}(z,t) = -(t\log(z) + (1-t)\log(1-z)) \tag{3}$$

where x obeys the high-dimensional global feature vector distribution of partial point clouds, y obeys the high-dimensional global feature vector distribution of ground truth, L_{bce} is binary cross-entropy loss function, z represents the network prediction output, and t donates the label of 0 or 1.

The training of generator G also needs to fix the parameters of discriminator D, and the loss function $\text{Loss}_{\text{adv}}^{G}$ is as follows:

$$\text{Loss}_{\text{adv}}^{G} = L_{\text{bce}}(D(x, G(x)), 1). \tag{4}$$

Minimizing this loss means that generator G will make discriminator D as "confused" as possible, that is, D cannot correctly distinguish the source of sample. However, in the actual training process, only minimizing this loss may lead to instability. We apply the $L1$ distance between y and GFV $G(x)$ generated by G into the loss function. The $L1$ loss function Loss_{L1} and the final loss function Loss_{G} are given as follows:

$$\text{Loss}_{L1} = \|y - G(x)\|_1 \tag{5}$$

$$\text{Loss}_{G} = \alpha\text{Loss}_{\text{adv}}^{G} + \beta\text{Loss}_{L1} \tag{6}$$

where α is the parameter of $\text{Loss}_{\text{adv}}^{G}$ in the loss function and β donates the parameter of Loss_{L1}. The task of generator G is not only to "deceive" D but also to be as close to target high-dimensional global feature vector y as possible.

IV. Experiment

In this section, we first introduce datasets and metrics and conduct extensive experiments on the ShapeNet and KITTI [40] datasets. Extensive experiments illustrate that the great effectiveness and generality of DPCG-Net outperform other methods with a large percentage of missing point clouds and it is equally valid for real point clouds. In addition, we design an ablation study to explore the design of different modules of DPCG-Net. Finally, we explore the application of point cloud completion in classification, where our method achieves consistent and significant performance improvements.

A. Dataset and Metric

1) Dataset: The ShapeNet dataset comes from PCN, which contains 30 974 models from eight categories, including airplane, cabinet, car, chair, lamp, sofa, table, and vessel. The complete point clouds contain 16 384 uniformly sampled point clouds. For each model, we obtain 2.5-D depth images from eight different random angles and utilize backprojected depth images to generate partial inputs. We split 100 models as the validation set, 150 models as the test set, and the rest as the training set.

The KITTI dataset is also from PCN, which is only used for the testing phase. A series of real point clouds collected with professional LiDAR scanning equipment is selected from the KITTI dataset. For each data frame, the point clouds under the car labels are extracted, and finally, a total of 2483 scenes of scanned car data are obtained. The LiDAR scanned car point clouds are very sparse, with only 440 points on average.

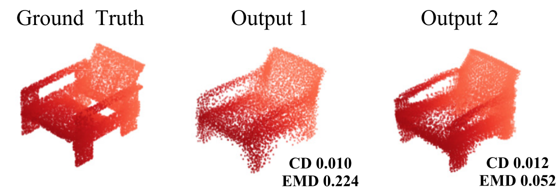


Fig. 4. Comparison of CD and EMD metrics. EMD is more reliable in distinguishing the visual quality of results.

2) Implementation Detail: The network is trained on a single NVIDIA GeForce RTX 2080 Ti. The encoder–decoder model is pretrained for 50 epochs with a batch size of 24. The value of α in the loss function Loss_{E-D} varies with the number of training steps from 0.01 to 0.1, 0.5, and 1.0 at 10 000, 20 000, and 50 000 steps, respectively. During the training stage of GAN, it is worth noting that the parameters of encoder and decoder remain unchanged and the parameters of generator and discriminator are updated iteratively. To make the numerical magnitudes of $\text{Loss}_{\text{adv}}^{G}$ and Loss_{L1} in the loss function Loss_{G} relatively close, this experiment chooses α as 0.1 and β as 20. The Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ was used for training. The update frequency of the parameters of discriminator and generator is 5:1. The learning rate of both generator and discriminator is set to 0.0001, and the batch size is set to 32 for a total of 120 training epochs.

3) Metric: We provide a detailed comparison of EMD and CD metrics. The formulas for calculating distance between two point sets P_1 and P_2 are described as follows:

$$\text{EMD}(P_1, P_2) = \min_{\phi: P_1 \to P_2} \frac{1}{|P_1|} \sum_{x \in P_1} \|x - \phi(x)\|_2 \tag{7}$$

$$\text{CD}(P_1, P_2) = \frac{1}{|P_1|} \sum_{x \in P_1} \min_{y \in P_2} \|x - y\|_2 + \frac{1}{|P_2|} \sum_{y \in P_2} \min_{x \in P_1} \|y - x\|_2. \tag{8}$$

The calculation of EMD is time-consuming, but it can provide a better representation of density distribution of point clouds, while CD is relatively simple to calculate but is more sensitive to outliers [30], [41]. Fig. 4 shows the performance of two different outputs on the CD and EMD metrics. Results are similar on the CD metric, but there is a significant difference between two outputs on the EMD metric, with output 2 (0.0523) outperforming output 1 (0.2241) by a significant margin. Also, as can be seen from a visualization perspective, Output 2 performs better, which is consistent with the conclusions drawn on the EMD metric, indicating that the EMD metric is more reliable relative to the CD metric in distinguishing the visual quality of results, in line with the conclusions of work [30]. Therefore, EMD is used as the main evaluation metric in this article.

In addition, since the data obtained by KITTI have no ground truth for reference, it cannot be directly quantified to indicate the completion and the registration error is used to measure the result. The rotation error R_e and the translation

TABLE I
EMD ON SHAPENET

Category	chair	car	cabinet	airplane	sofa	lamp	table	vessel
FC	1.897	1.573	2.147	0.736	1.915	2.406	1.611	1.418
Folding [24]	2.749	1.613	1.787	1.406	2.224	3.208	1.757	1.855
Topnet [16]	1.217	0.643	0.918	0.712	1.189	2.19	0.827	1.037
PCN [15]	0.681	0.550	0.707	0.388	0.724	0.846	0.601	0.627
N-DPC [35]	**0.638**	**0.531**	0.646	**0.385**	0.622	**0.841**	**0.572**	**0.605**
Ours	0.655	0.547	**0.611**	0.392	**0.597**	0.868	0.576	0.646

[1] Earth Mover's Distance (EMD) is scaled by 10.

error T_e are as follows:

$$R_e = 2\cos^{-1}(2\langle p_i, p_{gt}\rangle^2 - 1) \tag{9}$$

$$T_e = \left\|t_i - t_{gt}\right\|_2 \tag{10}$$

where p_i and q_{gt} are rotation computed by registration and ground truth rotation, respectively, t_i is the translation by registration, and t_{gt} is the ground truth translation.

B. Comparing Result

We compare our network with several state-of-the-art point cloud completion methods: FC, Folding, Topnet, PCN, and N-DPC. As the data used in the training phase of this part are the same as that in PCN, we use trained weights of FC, Folding, and PCN provided by PCN in Github. We retrain the weights of Topnet on ShapeNet dataset, set parameter ($L = 8$), and modify the output of last layer so that it outputs 16 384 dense point clouds. The weights of our previously proposed N-DPC have been trained in advance.

1) Test on ShapeNet: As shown in Table I, DPCG-Net outperforms the other models on the cabinet and sofa categories. In particular, the EMD error on the cabinet category is 0.611, which is 5% less than that of N-DPC model (0.646). The EMD error on the sofa category is 0.597, a 4% reduction from the error of N-DPC model (0.622). Experimental results on categories other than cabinet and sofa are comparable to the optimal N-DPC model, particularly in the airplane and table categories.

2) Masking Test: The above experiment indicates that the EMD performance of DPCG-Net in some categories obtained from backprojected 2.5-D depth images is inferior to the N-DPC model. In order to test the performance of trained model under different kinds of input data and further explore its completion performance, we test the performance of our model under different missing ratios of 30%, 50%, 60%, and 80% in comparison with other methods.

As shown in Table II, when the missing percentage of point clouds is 50%, 60%, and 80%, the error of DPCG-Net model is smaller than that of FC, Folding, Topnet, PCN, and N-DPC methods. Further analysis of the experimental results reveals that DPCG-Net gradually dominates as the proportion of deletions increases. In other words, DPCG-Net is weaker than N-DPC when 30% of point clouds are missing, and DPCG-Net leads the second N-DPC model by 2% at 50% missing rate. When the missing percentage increases to 60% and 80%, the errors of DPCG-Net compared to N-DPC model are reduced by 7% and 10%, respectively. It can be seen that the superiority of DPCG-Net model becomes more and more significant as the proportion of point clouds missing increases.

TABLE II
EMD ON MASKING TEST

Missing	30%	50%	60%	80%
FC	1.701	1.739	1.809	1.984
Folding [24]	2.016	2.176	2.325	2.531
Topnet [16]	1.018	1.178	1.325	1.606
PCN [15]	0.627	0.731	0.825	1.049
N-DPC [35]	**0.590**	0.665	0.747	0.936
Ours	0.611	**0.652**	**0.695**	**0.845**

[1] Earth Mover's Distance (EMD) is scaled by 10.

Due to the limitations of article layout and considering the specific performance of algorithm on different categories of ShapeNet, we only show the results of different models when 80% of point clouds are missing, as shown in Table III and Fig. 5. It can be inferred that DPCG-Net (0.845) has the smallest average value of EMD, followed by N-DPC (0.936), and Folding (2.531) has the largest error. In terms of specific categories, DPCG-Net has the best performance on all categories, including chairs, cars, and cabinets. Fig. 5 shows the visualization results of different methods when the missing ratio is 80%, which suggests that DPCG-Net is closer to ground truth than FC, Folding, Topnet, PCN, and N-DPC. Taking the chair category as an example, the overall results of FC and Folding models are rough, and the results of Topnet, PCN, N-DPC, and DPCG-Net models are more complete. However, DPCG-Net is more accurate in the details of armrests and legs of chair.

3) Future Discussion: Two types of data are used in the model testing phase of this article, as shown in Fig. 6. Fig. 6(a) shows the partial point clouds obtained by backprojection of 2.5-D depth images at random angles, with an average number of 1104 points for all samples. Fig. 6(b) shows the partial point clouds in different missing proportions. Taking the missing proportion of 80% as an example, partial point clouds are obtained by selecting a random point in ground truth and removing 16 384 × 80% of the proximity of that point from 16 384 point clouds. The former has a smaller number of point clouds, but it has a wider distribution of surface point clouds and contains richer structural information. Conversely, the latter has a larger number of points, but the distribution is

TABLE III
COMPLETION RESULTS ON THE EMD WITH A MISSING RATIO OF 80%

Category	chair	car	cabinet	airplane	sofa	lamp	table	vessel	mean
FC	2.309	1.635	2.476	0.994	2.048	2.759	1.918	1.733	1.984
Folding [24]	3.164	2.104	2.519	1.945	2.645	3.352	2.335	2.182	2.531
Topnet [16]	1.973	1.098	1.682	1.317	1.228	2.753	1.22	1.577	1.606
PCN [15]	1.246	0.701	1.178	0.706	0.836	1.69	1.011	1.028	1.049
N-DPC [35]	1.112	0.718	0.838	0.675	0.739	1.595	0.846	0.962	0.936
Ours	**0.982**	**0.636**	**0.668**	**0.595**	**0.665**	**1.553**	**0.787**	**0.871**	**0.845**

[1] Earth Mover's Distance (EMD) is scaled by 10.

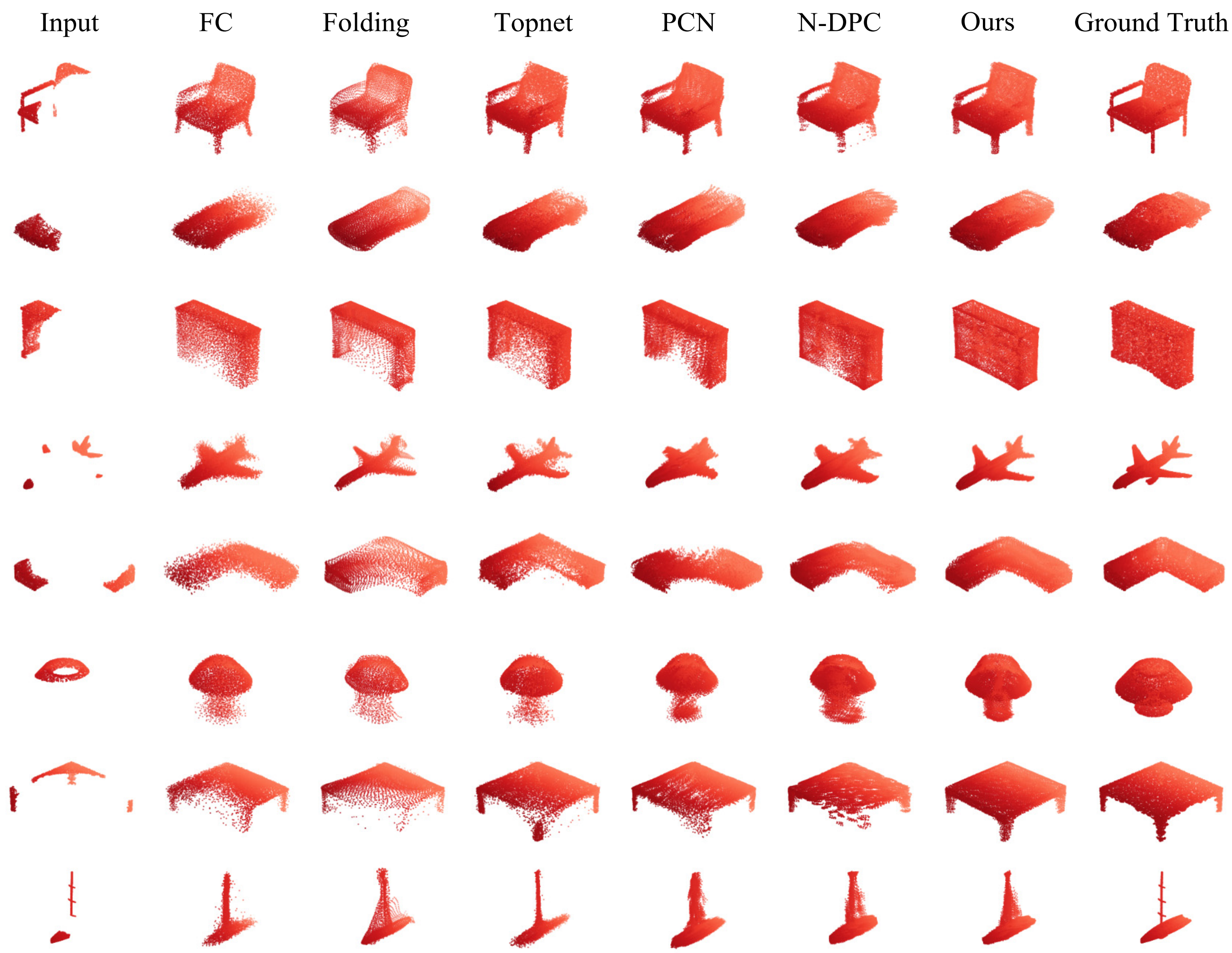


Fig. 5. Qualitative completion results at 80% missing on ShapeNet. Each output point clouds consists of 16 384 points.

more concentrated and only gathered in a small spatial area. The overall key structural information is less, which makes it relatively difficult to complete. Combined with the above experimental results, it demonstrates that DPCG-Net has a better performance in the case where a larger proportion of point clouds is missing, resulting in a larger overall structural deficiency in the data.

C. Test on KITTI

This section explores the model's complementary performance in realistic point clouds scenarios, with data from the KITTI car dataset in PCN. We use DPCG-Net trained on eight categories, including cars from the ShapeNet dataset to test the completion performance on the KITTI car dataset, the results of which are shown in Fig. 7. As scanned data do not have ground truth, Fig. 7(a) only shows the visualization of car before and after completion. The data after completion still retain the shape of car and are more informative, demonstrating the usefulness of DPCG-Net. We further investigate the performance of point clouds before and after completion in terms of registration using ICP. The results are shown in Fig. 7(b) and Table IV. The rotation and translation errors of point clouds are significantly

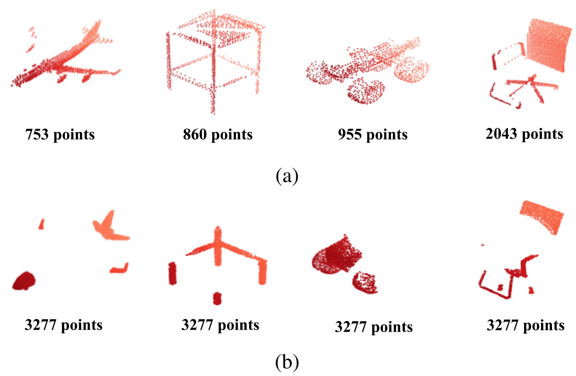


Fig. 6. Comparison of different test input data. (a) Backprojected depth images and (b) 80% missing of ground truth. The number below the image indicates the quantity of point clouds.

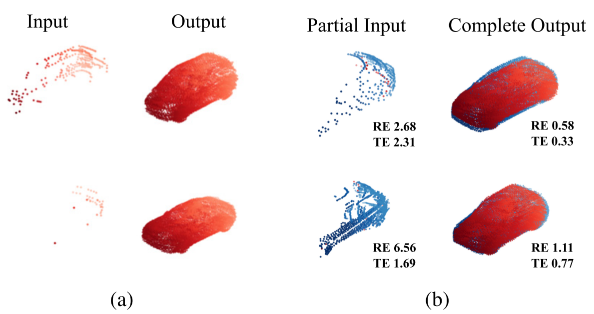


Fig. 7. Performance of (a) completion and (b) registration on KITTI car. In (b), RE indicates rotation error and TE donates translation error.

TABLE IV
REGISTRATION RESULTS ON KITTI CAR

Registration	Rotation error	Translation error
Partial input	14.09	7.11
Complete input	**7.36**	**4.88**

[1] Mean registration error with different inputs.

reduced after completion, reflecting the effectiveness of DPCG-Net.

D. Ablation Study

This section explores the influence of $L1$ distance in the loss function of generator and the setting of parameter d in discriminator on the experimental results. The loss function design of generator G in DPCG-Net applies the $L1$ distance between the high-dimensional global feature vector [GFV $G(x)$] generated by G and the high-dimensional global feature vector (GFV y) of ground truth. To explore its utility, $L1$ distance is removed from the loss function of G, and the experiment in which $L2$ replaces the $L1$ distance is also implemented. Under the same conditions, DPCG-Net is retrained and the results are shown in Table V. "No" means that the $L1$ distance is removed from the loss function, "With-$L1$" indicates the loss function used in the previous experiments, and "With-$L2$" denotes the loss function where $L2$ replaces $L1$. It can be noted that the error is minimum in the experiment for $L1$ distance. In particular, compared with directly removing the $L1$ distance, "With-$L1$" improves the performance of DPCG-Net by 11% on the input

TABLE V
ABLATION RESULT OF DISTANCE METRICS IN THE LOSS FUNCTION

EMD($\times$10)	No	With-$L1$	With-$L2$
Depth	0.685	**0.611**	0.632
80% Missing	1.070	**0.845**	0.881

[1] Depth represents the input from back-projected depth images. 80% Missing represents the input from 80% missing of ground truth.

TABLE VI
INFLUENCE OF THE CHANGE OF PARAMETER d

EMD($\times$ 10)	64	128	256
Depth	0.612(0.6121)	**0.611(0.6113)**	0.612(0.6119)
80% Missing	0.850	**0.845**	0.861

[1] Depth represents the input from back-projected depth images. 80% Missing represents the input from 80% missing of ground truth.

TABLE VII
EXPERIMENTAL RESULTS OF POINT CLOUDS CLASSIFICATION

Missing	30%	50%	60%	80%
Input	87.17%	70.75%	61.50%	44.67%
FC	94.50%	92.17%	90.33%	81.17%
Folding [24]	93.50%	91.67%	91.00%	83.67%
Topnet [16]	**95.00%**	**92.58%**	91.58%	84.25%
PCN [15]	94.67%	92.00%	90.67%	83.75%
N-DPC [35]	94.75%	92.25%	91.33%	84.33%
Ours	94.33%	92.17%	**92.00%**	**86.50%**

from backprojected depth images and by 21% under conditions where 80% of the ground truth is missing.

The discriminator in DPCG-Net divides the 1024-D global feature vector into small feature vectors of dimension d. To explore the parameter setting of d, we retrain the model with d of 64, 128, and 256. Also, the trained models are tested on different input data. The specific results are shown in Table VI. When d is 128, the model performs optimally but not significantly on the input from backprojected depth images. However, it performs significantly better than the other parameter settings on input from 80% missing of ground truth. Therefore, this article chooses $d = 128$ as the optimal parameter setting.

E. Application Into Classification

This section researches the application of dense point cloud completion into classification. A classifier network is designed into two parts: point feature extraction and point feature classification. Point features are extracted using shared MLPs with structure of [64, 128, 256, 1024], and a global feature vector of (1×1024) is obtained by the maxpool operation, which is then fed into fully connected layers of [512, 256, 8] for classification. The classifier is trained on the ground truth of ShapNet dataset.

We apply a trained classifier to test the classification results of partial point clouds under different missing ratios, as well as the point clouds completed with FC, Folding, Topnet, PCN, N-DPC, and DPCG-Net methods. The results are given in Table VII. In general, Topnet has the highest classification accuracy at 30% and 50% missing percentages. Nevertheless,

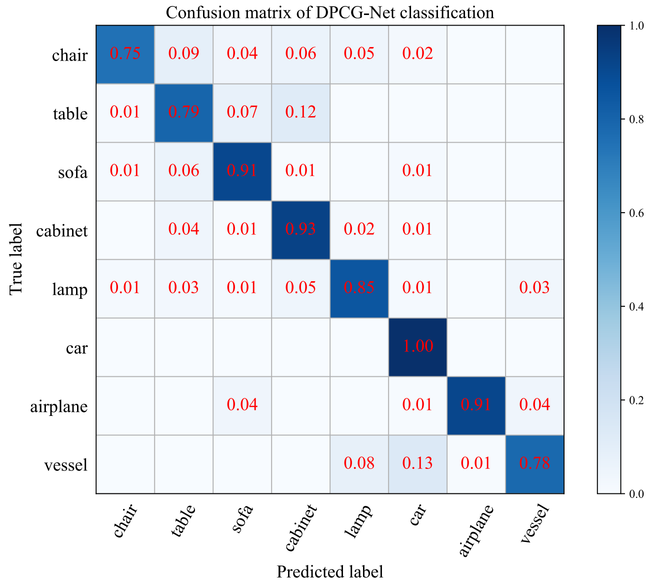


Fig. 8. Confusion matrix of the point clouds complemented by DPCG-Net in 80% missing condition.

when 60% and 80% of point clouds are missing, point clouds after the completion of DPCG-Net model have the best performance. The classification accuracy of DPCG-Net is as high as 86.5% when 80% of point clouds are missing, which leads N-DPC by 2%, demonstrating the superiority of our model under the condition of a large missing point cloud.

In addition, a confusion matrix is used to analyze the performance of DPCG-Net in different categories of classification. As shown in Fig. 8, it is able to achieve over 90% prediction accuracy on four categories, namely cars, planes, cabinets, and sofas, especially cars, 85% on lamps, and only about 78% on vessels, chairs, and tables. Further analysis of the confusion matrix suggests that 13% of vessels are misclassified as cars and 8% as lamps; 9% of chairs are mistakenly categorized as tables and 1% of tables are falsely labeled as chairs, which usually have a four-legged character and are difficult for classifier to distinguish; and 12% of tables are misclassified as cabinets and 4% of cabinets are incorrectly marked as tables, both of which have similar shape information and are mutually misclassified.

V. Conclusion

In this article, we propose a novel dense point cloud completion method combined with GAN, called DPCG-Net. Quantitative and qualitative evaluations of experiments demonstrate that our method performs best in the case of large percentage of missing point clouds resulting in large loss of spatial structure. KITTI results show that our method is also valid for real point clouds. In addition, the experimental results of point clouds classification validate the superiority of DPCG-Net in the case of large-scale missing point clouds. The model in this article primarily utilizes fully connected layers to extract features. Also, in the future, we will consider a combination of point- and voxel-based approaches for extracting features to further reduce the number of parameters in the algorithmic framework.

References

[1] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, Sep. 2019.

[2] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7662–7670.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[6] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*. [Online]. Available: http://arxiv.org/abs/1512.03012

[7] Z. Wu *et al.*, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[8] D. T. Nguyen, B.-S. Hua, M.-K. Tran, Q.-H. Pham, and S.-K. Yeung, "A field model for repairing 3D shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5676–5684.

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

[10] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive O-CNN: A patch-based deep representation of 3D shapes," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–11, 2018.

[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 5099–5108.

[13] X. Wen, T. Li, Z. Han, and Y.-S. Liu, "Point cloud completion by skip-attention network with hierarchical folding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1939–1948.

[14] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5898–5907.

[15] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 728–737.

[16] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 383–392.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1125–1134.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervent (MICCAI)*, 2015, pp. 234–241.

[19] A. Dai, C. R. Qi, and M. Niessner, "Shape completion using 3D-encoder-predictor CNNs and shape synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5868–5877.

[20] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 365–381.

[21] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3577–3586.

[22] P.-S. Wang, Y. Liu, and X. Tong, "Deep octree-based CNNs with output-guided skip connections for 3D shape and scene completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 266–267.

[23] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[24] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 206–215.

[25] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," 2017, *arXiv:1707.02392*. [Online]. Available: http://arxiv.org/abs/1707.02392

[26] S. Gurumurthy and S. Agrawal, "High fidelity semantic shape completion for point clouds using latent optimization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1099–1108.

[27] X. Chen, B. Chen, and N. J. Mitra, "Unpaired point cloud completion on real scans using adversarial training," 2019, *arXiv:1904.00069*. [Online]. Available: http://arxiv.org/abs/1904.00069

[28] Y. Peng *et al.*, "Sparse-to-dense multi-encoder shape completion of unstructured point cloud," *IEEE Access*, vol. 8, pp. 30969–30978, 2020.

[29] W. Zhang, C. Long, Q. Yan, A. L. H. Chow, and C. Xiao, "Multi-stage point completion network with critical set supervision," *Comput. Aided Geometric Des.*, vol. 82, Oct. 2020, Art. no. 101925.

[30] M. Liu, L. Sheng, S. Yang, J. Shao, and S.-M. Hu, "Morphing and sampling network for dense point cloud completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 11596–11603.

[31] X. Wang, M. H. Ang, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3043–3051.

[32] W. Yan, R. Zhang, J. Wang, S. Liu, T. H. Li, and G. Li, "Vaccine-style-net: Point cloud completion in implicit continuous function space," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2067–2075.

[33] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 512–528.

[34] H. Son and Y. M. Kim, "SAUM: Symmetry-aware upsampling module for consistent point cloud completion," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Nov. 2020, pp. 158–174.

[35] G. Li, Y. Chen, M. Cheng, C. Wang, and J. Li, "N-DPC: Dense 3D point cloud completion based on improved multi-stage network," in *Proc. 9th Int. Conf. Comput. Pattern Recognit.*, Oct. 2020, pp. 274–279.

[36] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 7354–7363.

[37] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: A point cloud upsampling adversarial network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7203–7212.

[38] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. CVPR*, Jul. 2017, pp. 605–613.

[39] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*. [Online]. Available: http://arxiv.org/abs/1511.05440

[40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[41] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3D reconstruction networks learn?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3405–3414.

Ming Cheng (Member, IEEE) received the Ph.D. degree in biomedical engineering from Tsinghua University, Beijing, China, in 2004.

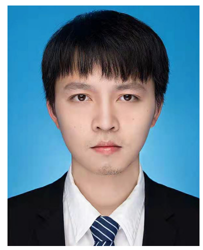
He is currently a Professor with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China. He has authored over 30 papers in refereed journals and conference proceedings, including IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *Neurocomputing*, and IEEE International Geoscience and Remote Sensing Symposium (IGARSS) and *International Society for Photogrammetry and Remote Sensing (ISPRS)* Proceedings. His research interests include remote sensing image processing, point cloud processing, computer vision, and machine learning.

Guoyan Li received the B.Eng. degree in computer science and technology from the School of Information Science and Technology, Xiamen University, Xiamen, China, in 2018, and the M.Eng. degree in computer science and technology from Fujian Key Laboratory Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, in 2021.

Her research interests include point clouds processing, shape completion, and machine learning.

Yiping Chen (Senior Member, IEEE) received the Ph.D. degree in information and communications engineering from the National University of Defense Technology, Changsha, China, in 2011.

From 2007 to 2011, she was an Assistant Researcher with The Chinese University of Hong Kong, Hong Kong. She is currently a Senior Engineer with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China. Her research interests include image processing, mobile laser scanning data analysis, point cloud computer vision, and autonomous driving.



Jun Chen received the B.Eng. degree in computer science and technology from the School of Informatics, Xiamen University, Xiamen, China, in 2020, where he is currently pursuing the master's degree in electronic information with Fujian Key Laboratory Sensing and Computing for Smart Cities.

His research interests include computer vision, deep learning, and information geometry.

Cheng Wang (Senior Member, IEEE) received the Ph.D. degree in signal and information processing from the National University of Defense Technology, Changsha, China, in 2002.

He is currently a Professor with the School of Informatics and the Executive Director of Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen, China. He has coauthored more than 150 papers in refereed journals and top conferences, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *Pattern Recognition (PR)*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE Conference on Computer Vision and PR, the Association for the Advancement of Artificial Intelligence (AAAI), and *International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing*. His research interests include point clouds analysis, multisensor fusion, mobile mapping, and geospatial big data.

Dr. Wang is a fellow of the Institution of Engineering and Technology. He is also the Chair of the Working Group I/6 on Multi-Sensor Integration and Fusion of the International Society of Remote Sensing.

Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000.

He is currently a Professor with the Department of Geography and Environmental Management and cross-appointed with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada. He is also a Founding Member of the Waterloo Artificial Intelligence Institute. He has coauthored more than 500 publications, over 280 of which were published in refereed journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, *International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing*, and *Remote Sensing of Environment*. His research interests include artificial intelligence (AI) techniques for information extraction from light detection and ranging (LiDAR) point clouds and Earth observation images and their applications in geospatial mapping, transportation, and urban digital twins.

Dr. Li was a recipient of the Outstanding Achievement Award in Mobile Mapping Technology in 2019 for his pioneering contributions in developing and promoting mobile mapping technology and the ISPRS Samuel Gamble Award in 2020 for his significant contributions to point cloud analytics in mobile LiDAR mapping. He is also the Chair of the ISPRS WG I/2 on LiDAR, Air- and Space-borne Optical Sensing from 2016 to 2022 and the ICA Commission on Sensor-Driven Mapping from 2015 to 2023. He is also the Editor-in-Chief of the *International Journal of Applied Earth Observation and Geoinformation*, an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and *Canadian Journal of Remote Sensing*.