IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 April 2023, accepted 31 May 2023, date of publication 7 June 2023, date of current version 20 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283920

RESEARCH ARTICLE

An Improved Point Cloud Completion Method Based on SnowflakeNet

MING CHEN, JINMING ZHANG, JIANLIANG LI, AND XIAOHAI ZHANG
School of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China
Corresponding author: Jinming Zhang (zhjmpt@163.com)

This work was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2022D01C690.

ABSTRACT Point cloud completion aims to complete partial point clouds captured from the real world, which is a crucial step in the pipeline of many point cloud tasks. Among the existing methods for solving this problem, SnowflakeNet is the most outstanding. However, SnowflakeNet cannot recover the detailed structure of point clouds in latent code because it uses many max-pooling operations in the encoding stage. Therefore, we propose an improved architecture to effectively acquire and preserve more detail information from input point clouds, thereby enhancing the quality of point cloud completion. Specifically, the improved lightweight DGCNN is added to the encoder to extract local features. The geometric perception block of PoinTr is introduced to extract the global features of the point cloud, which can fully model the structural information and inter-point relationships of known point clouds. The new optimizer Adan is also used in the training process to complete the partial point clouds. Comparative experiments on Completion3D and PCN datasets show that our method is better than most current point cloud completion methods. Our method has the ability to produce the entire shape with details, including but not only smooth surfaces, well-defined edges, and distinct corners.

INDEX TERMS Point cloud completion, feature extraction, improved lightweight DGCNN.

I. INTRODUCTION

Point clouds have driven the development of computer vision [1], [2], [3], [4], [5] as a commonly used, easily accessible data format that requires little memory to store and can convey detailed 3D shape information. Devices for capturing point clouds have become increasingly advanced provoking a significant amount of research (e.g., robotics, autonomous driving, and manufacturing). However, the point cloud data captured directly by these devices are often incomplete in realistic scenarios because of occlusion, reflection, and limitations in device resolution and angle. Therefore, the completion of such missing point clouds to obtain high-quality point clouds is crucial to facilitate downstream applications.

Recently, researchers have attempted to solve this problem. Early attempts at point cloud completion [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] via voxelization and 3D convolution. However, the amount of computation and the cost of these methods in processing point clouds are very large. With the success of PointNet and PointNet++ [17], [18], the 2D convolutional neural network that directly processes 3D point cloud itself has become mainstream. Currently, many tasks related to point cloud completion rely on PointNet and PointNet++ [17], [18]. The primary method involves extracting global features from the input point cloud using an encoder, which is subsequently decoded to generate a completion point cloud. However, these methods cannot recover the detailed structure of point clouds in code because it uses many max-pooling operations in the encoding stage. Therefore, capturing regional geometric details and structural characteristics(e.g., smooth surfaces and distinct borders), remains a challenging task in completing partial 3D shapes, as depicted in Fig. 1. (see Fig. 1(b) and Fig. 1(c)).

To address this issue, we propose an improved network structure that concentrates on improving input point cloud features extraction during the encoder phase. First, we propose an improved lightweight DGCNN for local region feature extraction in the encoder stage. DGCNN [21] is a dynamic graph-based CNN, which mainly addresses the problem of PointNet and PointNet++ [17], [18] only focus on point features and does not consider the geometric structure relationship in the local information. We further

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

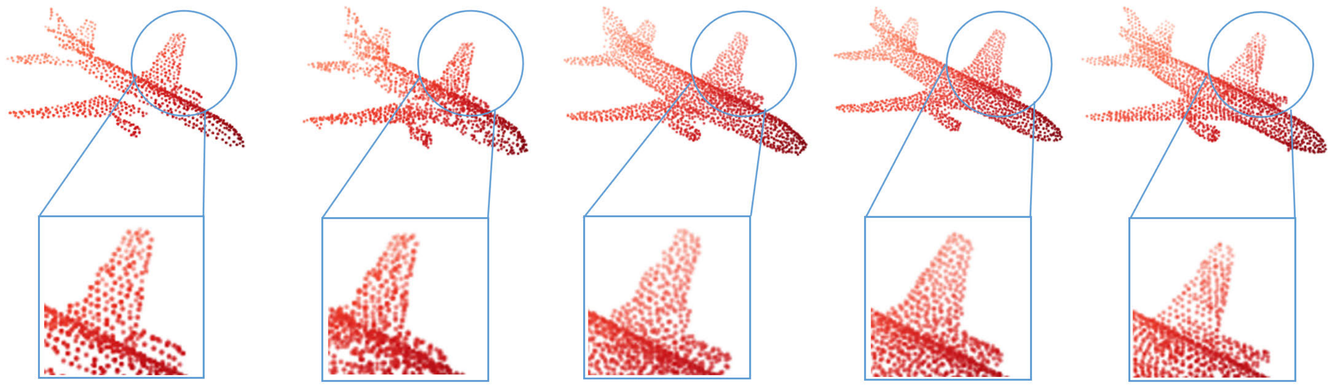

FIGURE 1. A comparative analysis of completed point cloud results through visual inspection. In contrast to PMP-Net [19] and SnowflakeNet [20], our network has the ability to produce the entire shape with details, including but not only smooth surfaces (blue box), well-defined edges, and distinct corners.

incorporate transformer structure into original DGCNN to enhance the capabilities of detail feature preservation. Second, we introduce the geometric perception block of PoinTr [22] to fully model the structural information and inter-point relationships of known point clouds. The output of the geometric perception block is used as the input of SnowflakeNet's decoder [20] to generate complete point clouds. Third, we also adopt a new deep model optimizer called Adan during training process to make the point cloud complete more effectively. Experiments on many public datasets show that our network architecture is better than most existing point cloud completion networks.

The main contributions of this paper are summarized as follows:

- The encoder is improved to better retain the fine-grained information of the input point cloud, which is conducive to the generation of complete point clouds at the decoder stage;
- An improved lightweight DGCNN is proposed for local feature extraction, which not only pays attention to the features of points, but also integrates the relationships between points in point cloud processing;
- To solve the shortage of SnowflakeNet in extracting global skeleton structure information from incomplete point clouds, inspired by poinTr, we introduce a transformer-based geometry-aware module to solve this problem.

II. RELATED WORK

A. POINT CLOUD COMPLETION BASED ON VOXEL

2D convolutional neural network has achieved great success in the application of plane image inpainting and restoration. Therefore, an intuitive idea for 3D shape completion is to build directly on the success of the 2D CNNs. Early methods for completing point clouds [6], [7], [9] attempted to transfer mature methods from 2D completion tasks to 3D point clouds using voxel positioning and 3D convolution. However, with an increase in spatial resolution, the computational cost of these methods is very high.

B. 3D SHAPE COMPLETION BASED ON ENCODER–DECODER

PointNet [17] is the first outstanding model that uses encoder-decoder architecture for point cloud processing tasks, which mainly addresses how to directly process 3D point clouds themselves using 2D CNNs. It can extract point set features stably even if the point clouds are fluctuating, noisy, or missing. However, PointNet [17] cannot effectively extract local fine features. To overcome this problem, PointNet++ [18] was proposed. PCN [23] directly processes the original point cloud without any structural assumptions (e.g., symmetry) or annotations of the underlying shape (e.g., semantic classes) and has a decoder design that allows the generation of fine-grained completions while maintaining a small number of parameters. DGCNN [21] addresses the problem that previous works focused only on point features and ignored the relationships between points, which employs EdgeConv to incorporate the relationship between points to build a dynamically updated graph model. In GRNet [24], grid and grid inversion methods are designed to convert point clouds into 3D grids, and a cubic feature sampling layer is proposed to extract information about neighboring points and preserve contextual knowledge. GRNet [24] allows convolution on 3D point clouds while preserving their structure and contextual information. PoinTr [22] considers point cloud completion as a set-to-set transformation problem, and proposes a transformer for point cloud completion. The point cloud is first transformed into a series of point agents, and then the transformer performs the point cloud generation. To facilitate the transformer to better exploit the sensing bias of the 3D geometric structure of the point cloud, they further design a geometry-aware block that explicitly models the local geometric relationships. SnowflakeNet [20] uses snowflake point deconvolution (SPD) and applies a transformer-based structure to the decoding process. SnowflakeNet [20] models complete point cloud generation as a snowflake-like growth of points in 3D space. After each SPD, child points are gradually generated by splitting their parent points.

III. METHODS

A. ARCHITECTURE OVERVIEW

The overall architecture of our network is shown in Fig. 2. We introduce our method in detail as follows.

1) FEATURE EXTRACTION MODULE

The purpose of feature extraction is to obtain a 1xC shape code, which captures the characteristics of input point cloud and relationships between points. To achieve this, we first improve the DGCNN [21], and then use the improved lightweight DGCNN in Fig. 5 (a) to extract the features of input point cloud. Finally, the global feature of input point cloud is obtained using the geometric perception transformer.

2) SEED GENERATION MODULE

The Seed generation module is borrowed from SnowflakeNet. The purpose of the seed generation module in Fig. 3 is to generate a coarse point cloud P_0 of size N_0x3. It is a complete point cloud that includes point cloud features, structure, and relationships between points. A coarse point cloud is used as the input of the point generation module in Fig. 2 to generate fine-grained point clouds.

The details of the Seed generation module are presented in Fig. 3. First, the global feature f from the feature extraction module is used as the input of the seed generation module, and then point features pf are generated through ConvTranspose1d. Second, a coarse point cloud is obtained by integrating f and pf using multi-layer perceptrons. Third, to make a coarse point cloud contain more prior information, we further concatenate P_c and the input incomplete point cloud P, and then we conduct down-sampling of its results to P_0 through FPS.

3) POINT GENERATION MODULE

In the decoding stage, we use the transformer-based SPD module of SnowflakeNet [20] in Fig. 3. The SPD is a deconvolution network with multiple layers. Each SPD layer uses the previous point cloud as its input and splits every parent point into several child points. This kind of generation process is like snowflakes growing in 3D space. To better restore local detail features during point splitting, each SPD layer is equipped with Skip-Transformers to capture regional shape characteristics and standardize the splitting patterns between adjacent SPDs. This approach not only allows point splitting to conform to local features but also enables neighboring SPDs to cooperate with each other, which ensures consistency in multi-step point splitting.

The details of SPD are shown in Fig. 4. Within the SPD, we first obtain each point feature in P_{i-1} using PointNet [17] (PN). Second, we concatenate each point feature and the global feature. Third, we use a Skip-Transformer (ST) to integrate the local shape context information and the displacement features of the previous SPD step. Then, we split each parent point feature by ConvTranspose1d to generate child point features, and the child point feature uses MLP to generate the displacement feature and the displacement vector of the child points. The displacement feature is introduced into the next SPD step to guide the next splitting step; The displacement vector ΔP_i represents the displacement of a child point relative to its parent point. Finally, upsample point P_{i-1} and add its result to ΔP_i to obtain P_i.

B. THE IMPROVED LIGHTWEIGHT DGCNN

Among the existing methods, the fine-grained details of input point cloud are easily lost during the pooling operation in the encoding phase. It is difficult to recover from diluted global features in the generation. To capture many more features of input point cloud, we propose an improved lightweight DGCNN to enhance feature extraction ability. First, the Transformer in Fig. 5 (b) is added to the DGCNN for feature extraction. Second, each extracted feature is passed through skip-concatenation, as shown in Fig. 5 (a). Module1 includes three steps: get graph feature, conv2d, and transformer. Module2 includes two steps: get graph feature, conv2d.

The details of the improved lightweight DGCNN are shown in Fig. 5. First, input a point cloud P with a size of Nx3. Second, we use the Input transformer module to turn each point of the input point cloud into an 8-dimensional coordinate. Third, we use module1 and max pooling operation to obtain features f1 and f2 of the input point cloud. Then, we use farthest point sampling (fps) to decrease the number of points of the input point cloud N_1, N_2. Next, we obtain the input point cloud features f3 using module2_1 and max pooling operation. Fifth, we concatenate f1, f2, and f3 using skip-concatenation. Then, the dimension of concatenated features is increased and max pooling is applied to obtain the feature f4. Finally, f1, f2, f3, and f4 are concatenated to form the input point cloud features. *Coor* and *f* represent the output results of points and features, respectively.

C. THE GEOMETRIC PERCEPTION TRANSFORMER BLOCK

Inspired by PoinTr, we also propose a geometry-aware block, as shown in Fig. 6 to fully model the structural information of known points and relationships between points so as to reduce information loss of input point cloud. First, we process the point *coor* and feature *f* obtained from the improved lightweight DGCNN: (1) capture the geometric relation *knn* in the *coor* by knn_index; (2) encode the global position of the *coor* by position embedding to obtain *pos*; (3) reduce dimension of the feature *f* and represent the result in x. Second, the geometric features and semantic features are obtained by the KNN query and Self-Attention. Third, the geometric features and semantic features are concatenated, and then, the global features f of the input point cloud are obtained by increasing dimension and pooling operation to the result.

D. TRAINING LOSS

In the experiment, the model is optimized using the chamfer distance (CD). To constrain the point clouds produced during the decoder stage, we downsample the ground-truth point cloud with the same sample density as P_0, P_1, P_2 and P_3 in Fig. 2. The predicted point clouds with the same density are grouped two by two with the ground-truth point cloud, the average distance of the nearest points between each group is calculated by CD, and the completion loss is derived by aggregating the losses of the four CDs, denoted by L_{com}.

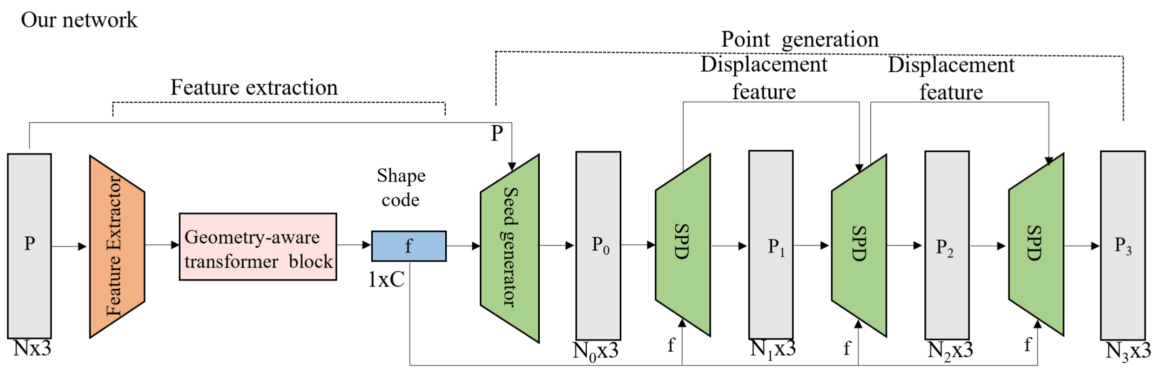

FIGURE 2. The overall architecture of network. We first extract local features from input point cloud by the improved lightweight DGCNN, and then we get the global features by the geometry-aware module. Then seed generator generates coarse point clouds, and finally, fine-grained point clouds are generated by Multiple SPDs and Skip-Concatenations.

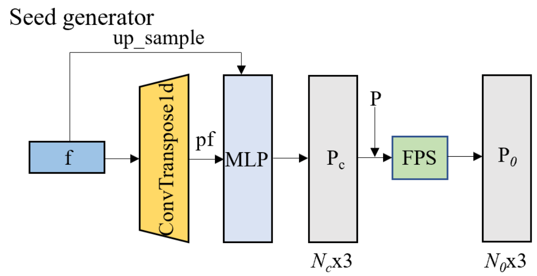

FIGURE 3. The overall of the Seed generation module.

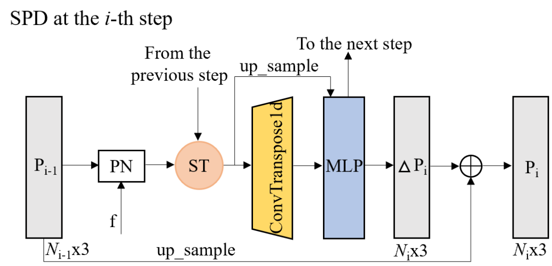

FIGURE 4. The overall of the SPD (Snowflake Point Deconvolution).

Moreover, we introduce L_{pre} loss to ensure that the shape structure of the input point cloud is retained. The final loss expression is as follows:

$$L_{com} = CD_0 + CD_1 + CD_2 + CD_3 \tag{1}$$

$$L = L_{com} + aL_{pre}. \tag{2}$$

IV. EXPERIMENTS

To demonstrate the effectiveness of our network, we evaluated it on PCN and Completion3D datasets, respectively. In point cloud completion research, our experiments show that our network is better than most existing networks available.

A. EVALUATION ON PCN DATASET

The PCN [23] dataset for dense point cloud completion, a synthetic CAD model of ShapeNet [25], is used to create a large scale dataset containing paired local and complete point clouds (X, Y). The dataset contains eight categories: airplanes, cabinets, cars, chairs, lamps, sofas, tables, and vessels. The complete point cloud was created by sampling 16384 points uniformly from the mesh surface, and the partial point cloud was generated by back-projecting the 2.5D depth map to 3D. To demonstrate the effectiveness of the network, in point cloud completion study methods, our network is compared with other networks on the PCN dataset, where L1 chamfer distances are used for evaluation. In order to ensure a fair comparison on the PCN dataset, our experimental configuration is identical to that of prior methods.

1) QUANTITATIVE COMPARISON

The performance of our network and other point cloud completion methods on the PCN dataset are presented in Table 1. Our method exhibits the highest performance among all evaluated methods. (Lld refers to the network that learning local displacements for the point cloud completion)

2) VISUAL COMPARISON

We select three typical point cloud completion methods(PMPNet [19], PMP-Net++ [31], and SnowflakeNet [20]) from Table 1, and illustrate a visual comparison between our network and the methods in Fig. 7. Reconstructed point clouds of PMP-Net and PMP-Net++ have 2048 points, other methods have 16384 points. It can be inferred from the visual results that our network is better at completing the partial point cloud. For example, in the cabinet category, the cabinet generated by our network with smooth surfaces, well-defined edges, and distinct corners.

B. EVALUATION ON COMPLETION3D DATASET

The Completion3D dataset is utilized for completing sparse point clouds, with its complete point cloud containing 2048 points. Its origin can be traced back to the ShapeNet [25] dataset. It is an online platform used to evaluate shape completion methods. Both the input and ground-truth point clouds have a resolution of 2048 points. Therefore,

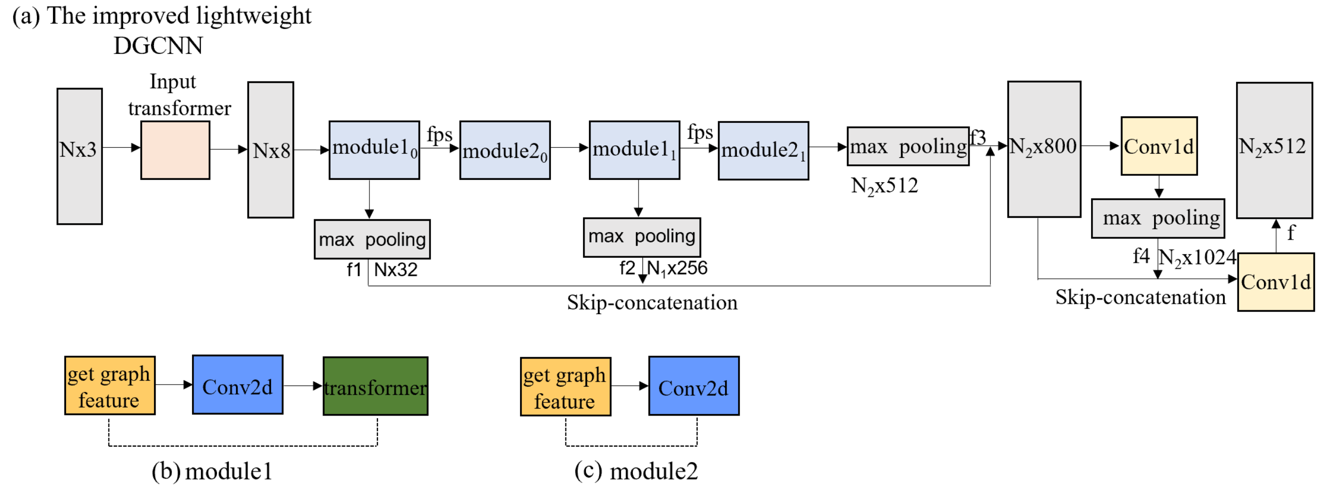

FIGURE 5. (a) Shows the overall of the improved lightweight DGCNN; (b) and (c) are components of the improved lightweight DGCNN.

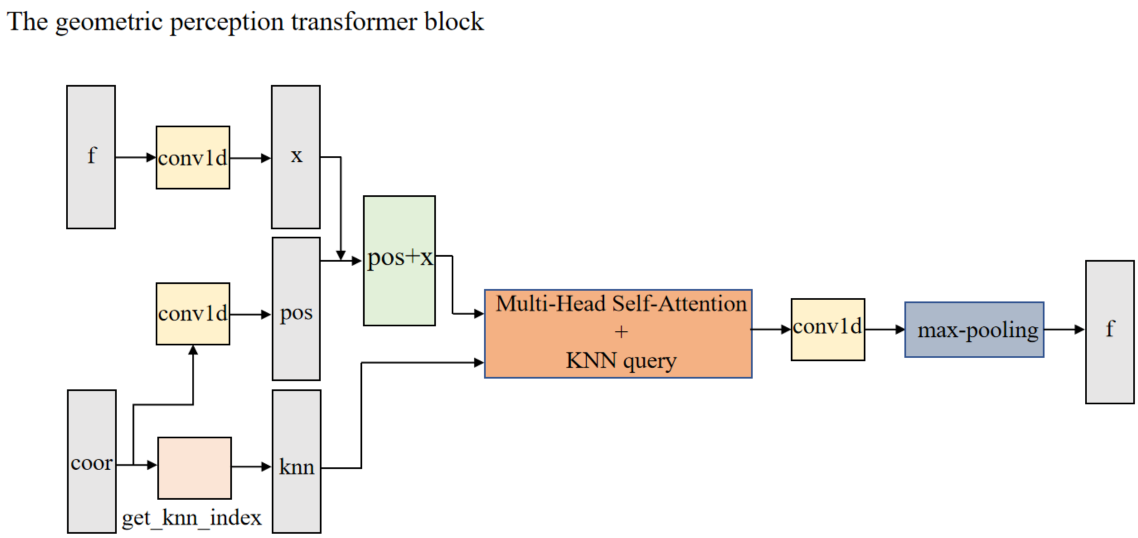

FIGURE 6. The geometric perception transformer block.

TABLE 1. Point cloud completion on PCN dataset in terms of per-point L1 Chamfer distance × 1000 (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
FoldingNet [26]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet [27]	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet [28]	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
PCN [23]	9.64	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59
GRNet [24]	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
CRN [29]	8.51	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05
NSFA [30]	8.06	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48
PMP-Net [19]	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
PoinTr [22]	8.38	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
PMP-Net++ [31]	7.56	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52
SoftPool++ [32]	8.31	5.50	10.02	8.73	9.05	7.53	10.24	8.01	7.43
Snowflake [20], [33]	7.21	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
Lld [34]	7.96	-	-	-	-	-	-	-	-
Ours	7.03	3.81	8.93	8.05	7.17	6.55	8.32	7.12	6.34

L2 chamfer distances are also used for evaluation on the Completion3D dataset. To ensure fairness, the experimental configuration is identical to that of the prior methods.

1) QUANTITATIVE COMPARISON

The performance of our network and other point cloud completion methods on the Completion3D dataset are presented in Table 2. Our method outperformed all evaluated methods.

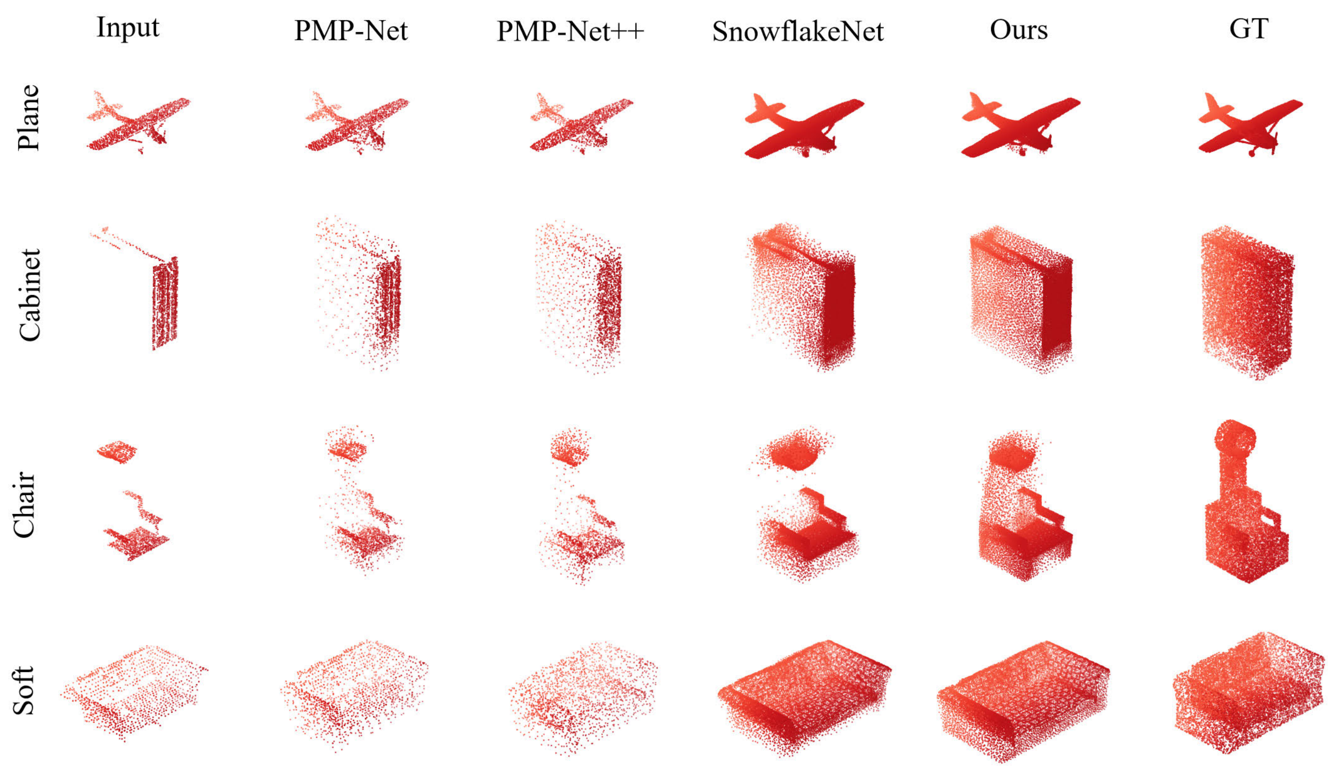

FIGURE 7. Visual comparison of PMP-Net, PMP-Net++, SnowflakeNet and our network on PCN dataset. The point clouds complete by our network have smooth surfaces, well-defined edges, and distinct corners.

TABLE 2. Point cloud completion on Completion3D dataset in terms of per-point L2 Chamfer distance × 10000 (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
FoldingNet [26]	19.07	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51
TopNet [27]	14.25	7.32	18.77	12.88	19.82	14.60	16.29	14.89	8.82
PCN [23]	18.22	9.79	22.70	12.43	25.14	22.72	20.26	20.27	11.73
AtlasNet [28]	17.77	10.36	23.40	13.40	24.16	20.24	20.82	17.52	11.62
GRNet [24]	10.64	6.13	16.90	8.27	12.23	10.22	14.93	10.08	5.86
CRN [29]	9.21	3.38	13.17	8.31	10.62	10.00	12.86	9.16	5.80
SA-Net [34]	11.22	5.27	14.45	7.78	13.67	13.53	12.22	11.75	8.84
PMP-Net [19]	9.23	3.99	14.70	8.55	10.21	9.27	12.43	8.51	5.77
PMP-Net++ [31]	7.97	3.25	12.25	7.62	8.71	7.62	11.6	7.06	5.38
SoftPool++ [32]	9.36	4.59	15.82	6.78	11.41	8.82	13.37	9.15	4.93
Snowflake [20], [33]	8.71	2.24	12.33	6.12	10.62	11.47	8.79	12.95	5.20
Ours	7.89	2.07	11.14	5.83	9.95	8.50	8.19	13.28	4.20

2) VISUAL COMPARISON

We selected three typical point cloud completion methods(PMPNet [19], PMP-Net++ [31], and SnowflakeNet [20]) from Table 2, and illustrate a visual comparison between our network and these methods. Reconstructed point clouds of all the methods have 2048 points. The visual results show our network is better at completing the partial point cloud, as shown in Fig. 8. For example, in the boat category, the boat is generated by our network with smooth surfaces, well-defined edges, and distinct corners.

C. ABLATION STUDIES

To prove the effectiveness of several architecture designs in our network, experiments are conduct ablation. The results are presented in Table 3. By default, all experimental configurations are identical, except for the analysis component. And all experiments are trained on the Completion3D dataset.

TABLE 3. Effect of each part in our designed network.

Methods	CD-Avg	Chair	Lamp	Vessel
SnowflakeNet-baseline	8.71	10.62	11.47	5.20
Without-the-improved-lightweight-DGCNN	7.99	9.58	8.57	4.69
Without-the-geometric-perception-block	7.99	10.20	8.32	4.40
Full	7.89	9.95	8.50	4.20

In order to analyze the effectiveness of the geometric perception block and the improved lightweight DGCNN separately, we develop two network variations as follows.

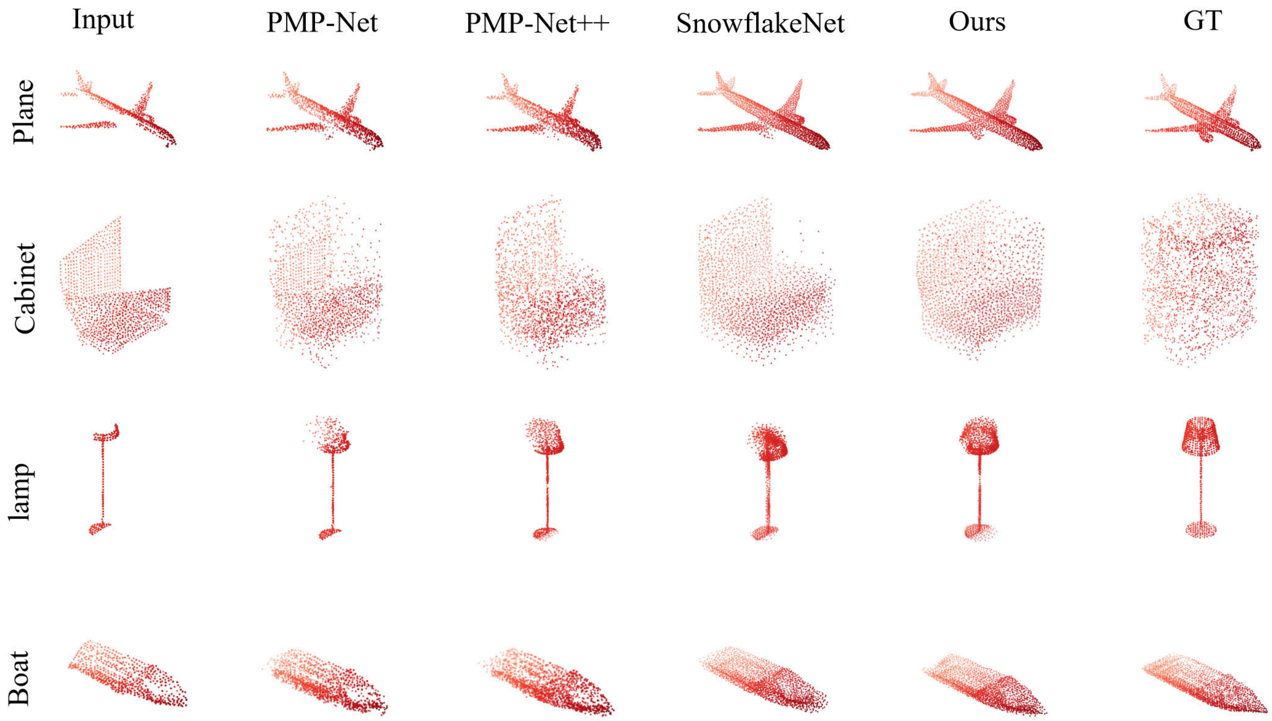

FIGURE 8. Visual comparison of PMP-Net, PMP-Net++, SnowflakeNet and our network on Completion3D dataset. Our network is superior to many other existing point cloud completion techniques in completing point clouds.

(1) Add the geometric perception block to SnowflakeNet for obtaining a global feature; (2) Use the improved lightweight DGCNN instead of Pointnet++'s [18] SA local feature extraction method to obtain local features of the input point cloud. Specially, transformers are added to DGCNN to obtain more local features. In addition, we mark Full as the complete network containing the geometric perception block and the improved lightweight DGCNN. It can be observed from Table 3 that the Full network surpasses all other analyzed network variations in terms of performance. The comparison between the Full model and the No-the-improved-lightweight-DGCNN model demonstrates the effectiveness of using the improved lightweight DGCNN, and the comparison between the Full model and the No-the-geometric-perception-block model justifies the advantage of using the geometric perception block.

V. CONCLUSION

In this paper, we propose an improved neural network for point cloud completion. By using the improved lightweight DGCNN to replace SA local feature extraction method, the local features extracted from the input point cloud not only focus on the point features but also incorporate the relationship between points into the point cloud processing. It is worth noting that further introduce transformers to DGCNN greatly improves the situation where fine-grained information is easily lost during the pooling processes at the encoder stage. Through experimental analysis, we demonstrate the superiority of our network by experiment comparisons with other networks on both the Completion3D dataset and the PCN dataset. In the future, I will further improve the completion effect of parint clouds by improving the SnowflakeNet decoder and enhance the feature extraction ability of the improved lightweight dgcnn.

REFERENCES

[1] Z. Han, C. Chen, Y.-S. Liu, and M. Zwicker, "ShapeCaptioner: Generative caption network for 3D shapes by learning a mapping from parts detected in multiple views to sentences," in *Proc. 28th ACM Int. Conf. Multimedia*, Seattle, WA, USA, Oct. 2020, pp. 1018–1027.

[2] Z. Han, M. Shang, X. Wang, Y. Liu, and M. Zwicker, "Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences," in *Proc. AAA*, Honolulu, HI, USA, 2019, pp. 126–133.

[3] Z. Han, M. Shang, Y. Liu, and M. Zwicker, "View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions," in *Proc. AAAI*, Honolulu, HI, USA, 2019, pp. 8376–8384.

[4] Z. Han, Z. Liu, C. Vong, Y. Liu, S. Bu, J. Han, and C. L. P. Chen, "BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3707–3720, Aug. 2017.

[5] Z. Han, Z. Liu, C. Vong, Y. Liu, S. Bu, J. Han, and C. L. P. Chen, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3049–3063, Jun. 2018.

[6] 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)*, Venice, Italy, Oct. 2017, pp. 85–93.

[7] 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)*, Honolulu, HI, USA, Jul. 2017, pp. 6545–6554.

[8] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proc. ECCV*, Amsterdam, The Netherlands, 2016, pp. 236–250.

[9] D. Stutz and A. Geiger, "Learning 3D shape completion from laser scan data with weak supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1955–1964.

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

[11] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 2442–2447.

[12] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. NIPS*, Vancouver, BC, Canada, 2019, pp. 963–973.

[13] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 8887–8896.

[14] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[15] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni, "3D object reconstruction from a single depth view with adversarial learning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 679–688.

[16] W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, "Shape inpainting using 3D generative adversarial network and recurrent convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2317–2325.

[17] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 77–85.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NIPS*, Long Beach, CA, USA, 2017, pp. 5099–5108.

[19] X. Wen, P. Xiang, Z. Han, Y. Cao, P. Wan, W. Zheng, and Y. Liu, "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7439–7448.

[20] P. Xiang, X. Wen, Y. Liu, Y. Cao, P. Wan, W. Zheng, and Z. Han, "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 5479–5489.

[21] A. V. Phan, M. L. Nguyen, Y. L. H. Nguyen, and L. T. Bui, "DGCNN: A convolutional neural network over large-scale labeled graphs," *Neural Netw.*, vol. 108, pp. 533–543, Dec. 2018.

[22] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 12478–12487.

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

[24] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Proc. ECCV*, Glasgow, U.K., 2020, pp. 365–381.

[25] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[26] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 206–215.

[27] 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)*, Long Beach, CA, USA, Jun. 2019, pp. 383–392.

[28] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "AtlasNet: A Papier-mâché approach to learning 3D surface generation," in *Proc. CVPR*, Salt Lake City, UT, USA, 2018, pp. 216–224.

[29] 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)*, Seattle, WA, USA, Jun. 2020, pp. 787–796.

[30] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *Proc. ECCV*, Glasgow, U.K., 2020, pp. 512–528.

[31] X. Wen, P. Xiang, Z. Han, Y. Cao, P. Wan, W. Zheng, and Y. Liu, "PMP-Net++: Point cloud completion by transformer-enhanced multi-step point moving paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 852–867, Jan. 2023.

[32] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "SoftPool++: An encoder–decoder network for point cloud completion," *Int. J. Comput. Vis.*, vol. 130, no. 5, pp. 1145–1164, May 2022.

[33] P. Xiang, X. Wen, Y. Liu, Y. Cao, P. Wan, W. Zheng, and Z. Han, "Snowflake point deconvolution for point cloud completion and generation with skip-transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 6320–6338, May 2023.

[34] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Learning local displacements for point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 1558–1567.

MING CHEN received the B.E. degree in computer science and technology from Minjiang University, Fuzhou, China, in 2020. She is currently pursuing the degree in computer technology with Xinjiang University, Ürümqi, China.

Her research interest includes computer vision. In 2020, she was awarded the title of Outstanding Graduate.

JINMING ZHANG received the B.S. and M.S. degrees in automation from Xi'an Jiaotong University, China, in 2011 and 2014, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, in 2019. He is currently an Associate Professor with the School of Information Science and Engineering, Xinjiang University. His major research interests include intelligent optimization algorithms, machine learning, and computer vision.

JIANLIANG LI received the B.E. degree in the Internet of Things from the Hunan University of Technology, China, in 2021. He is currently pursuing the degree in computer science and technology with Xinjiang University, Ürümqi, China.

His research interest includes computer vision.

XIAOHAI ZHANG received the B.E. degree in network engineering from the City Institute, Dalian University of Technology, China, in 2019. He is currently pursuing the degree in computer technology with Xinjiang University, Ürümqi, China.

His research interest includes computer vision.