



Article

Multistage Adaptive Point-Growth Network for Dense Point Cloud Completion

Ruidong Hao ^{1,2}, Zhonghui Wei ^{1,*}, Xu He ³, Kaifeng Zhu ^{1,2}, Jun Wang ¹, Jiawei He ¹ and Lei Zhang ¹

¹ Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

² University of Chinese Academy of Sciences, Beijing 100049, China

³ College of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China

* Correspondence: weizhonghui@ciomp.ac.cn

Abstract: The point cloud data from actual measurements are often sparse and incomplete, making it difficult to apply them directly to visual processing and 3D reconstruction. The point cloud completion task can predict missing parts based on a sparse and incomplete point cloud model. However, the disordered and unstructured characteristics of point clouds make it difficult for neural networks to obtain detailed spatial structures and topological relationships, resulting in a challenging point cloud completion task. Existing point cloud completion methods can only predict the rough geometry of the point cloud, but cannot accurately predict the local details. To address the shortcomings of existing point cloud complementation methods, this paper describes a novel network for adaptive point cloud growth, MAPGNet, which generates a sparse skeletal point cloud using the skeletal features in the composite encoder, and then adaptively grows the local point cloud in the spherical neighborhood of each point using the growth features to complement the details of the point cloud in two steps. In this paper, the Offset Transformer module is added in the process of complementation to enhance the contextual connection between point clouds. As a result, MAPGNet improves the quality of the generated point clouds and recovers more local detail information. Comparing our algorithm with other state-of-the-art algorithms in different datasets, experimental results show that our algorithm has advantages in dense point cloud completion.

Keywords: point cloud; dense point cloud completion; point growth; composite encoder; Offset Transformer



Citation: Hao, R.; Wei, Z.; He, X.; Zhu, K.; Wang, J.; He, J.; Zhang, L. Multistage Adaptive Point-Growth Network for Dense Point Cloud Completion. *Remote Sens.* **2022**, *14*, 5214. <https://doi.org/10.3390/rs14205214>

Academic Editors: Pawel Rotter, Wojciech Chmiel and Sławomir Mikrut

Received: 13 September 2022

Accepted: 14 October 2022

Published: 18 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Dense point cloud complementation is a task to estimate and predict a dense and complete point cloud based on a sparse and incomplete point cloud. With the development of remote sensing technology, LiDAR, depth cameras and other 3D scanning devices are applied to various industries, and they can easily acquire massive point cloud data [1–4]. The simple format, versatility and small memory footprint of point cloud data [5] are heavily used in 3D reconstruction, autopilots, virtual reality, map drawing and machine vision [6–10]. However, due to the resolution limitation of the sensor, the occlusion of other objects, the surface light reflection of the object itself, the manual error and the perspective problems, the quality of the original point cloud data will decrease. The low quality of the point cloud is reflected in the sparse distribution and the incomplete structure. A complete and dense point cloud is the fundamental condition for 3D computer vision applications such as point cloud classification, segmentation and other 3D analysis and evaluation methods [11–15]. In practical applications, a dense and complete point cloud helps to better reconstruct the 3D model of a physical object, which can be better applied to industry and production [16,17]. Therefore, it is of great theoretical significance and application value to

study the recovery of dense and complete point clouds from the observed incomplete point cloud data.

Incomplete, disordered and unstructured point cloud data make point cloud learning a challenge. Thanks to the development of deep learning, the method of point cloud learning has made great progress. With the successful application of the famous pioneer work PointNet [11] and its improved PointNet++ [18] to point cloud learning, various point cloud learning methods that rely on PointNet and PointNet++ as the encoder design are emerging [19,20]. Recently, this encoder–decoder architecture has also been successfully applied to the point cloud completion task, such as representative FoldingNet [21], PCN [22], TopNet [23], etc., which can extract the features of incomplete point clouds and predict the generation of complete point clouds. However, its encoder–decoder capability is limited, which cannot fully analyze and generate accurate point cloud models. The later developed GRNet [24], PFNet [25], PMP [26] and other algorithms further improved the generated performance indicators, but the results still have the problem of distortion. Especially when the structure of the point cloud is complex and the number of generated points is large, the existing completion algorithms may fail to depict the local details and smooth surfaces of the point cloud model, sometimes having rough edges and scattered points. In serious cases, there may even be some failure results. Therefore, it is still a very difficult task to use neural networks to accurately complete the dense point cloud.

In this paper, to solve the above problem, we propose a new neural network named MAPGNet, which generates complete and dense point clouds in a coarse-to-fine form step by step. Different from the existing point cloud completion algorithm, the network we designed focuses more on generating local details of dense point clouds. In the process of expanding point clouds after generating skeleton point clouds, the local detail point clouds are generated in a more free way. The proposed algorithm is divided into three modules: composite encoder feature extraction module, skeleton point cloud generation module and point cloud growth module. The feature extraction module of the composite encoder extracts three different stages of point cloud features, which are skeleton feature, growth feature 1 and growth feature 2. The three different features extracted are responsible for three different completion stages. The skeleton features extracted from the composite encoder generate sparse and complete skeleton point clouds using the skeleton point cloud generation module. Next, based on growth feature 1/2, local point cloud details are generated adaptively in the sphere neighborhood by using the point cloud growth module in two stages on the basis of the previous generated point cloud. The point cloud growth module combines the growth features with the Offset Transformer structure and the global features of the skeleton point cloud, which better correlates the topological relationship and context information of the point cloud. The two-stage point cloud growth module refines the local geometry information of the point cloud step by step and finally obtains a complete and dense point cloud model. We conducted experiments on the PCN dataset [22] and the Completion3D dataset [23] to analyze the effectiveness of the module and to demonstrate the state-of-the-art quantitative and visualization results of the method. The experimental results on different datasets show that our network outperforms existing algorithms in the dense point cloud completion task, demonstrating its superior performance. In all, the main contributions can be summarized as follows:

- A new point cloud completion network, MAPGNet, is proposed, which completes missing point clouds in a phased manner into complete dense high-quality point clouds in an adaptive point cloud growth manner. Compared with previous point cloud completion tasks, our method can preserve the details of the input missing point cloud and generate the missing point cloud parts with high quality.
- A composite encoder structure is proposed in which different encoding structures in the composite encoder can focus on different complementary phases. Different from the previous single encoder, the composite encoder with Offset Transformer fully extracts the global frame information, local detail information and context information

associated with the input missing point cloud, which further improves the ability of the point cloud completion task.

- The point cloud growth module proposed combines the features of the missing point cloud and complete skeleton point cloud to grow dense point clouds adaptively in the predetermined spherical neighborhood. The resultant point cloud surface is smoother and the edge is sharper.
- It is shown on different datasets that our neural network is superior to the existing algorithm in the dense point cloud completion task.

2. Related Work

Traditional Methods. Traditional point cloud completion methods are usually divided into those based on geometric structure information [27] and those based on template retrieval [28]. The method based on geometric structure usually reconstructs the surface of the point cloud manifold to repair the incomplete mesh or fills the hole by using the neighborhood information of the missing point cloud [29]. Template-based retrieval completes shape repair by matching missing point clouds from input to templates in the database. [30] The method based on geometric structure loses the topological relationship and cannot be applied to large-scale missing point clouds, while the method based on template retrieval requires a lot of prior knowledge and manual operations, and cannot complete the shape of unknown objects.

Learning Methods. With the booming development of deep neural networks, a lot of research has been carried out on related tasks such as point cloud classification, segmentation and recognition [11,18]. These tasks respectively attempt to represent the geometrical information of point cloud with high-dimensional features. Currently, learning-based point cloud completion methods mainly include multiview, voxel-based and direct point cloud-based methods.

Many early works projected point clouds onto 2D planes and used common 2D convolution operations to extract features of several new plane views [31,32], but this method completely ignored the spatial structure of point clouds and the complementary point clouds were not satisfactory.

In order to make the point cloud as uniform and ordered as images, the method of 3D voxelization of point clouds has been created. This method uses 3D convolution to convolve the cubic units after voxelization, such as 3D-ResGAN [33], 3D-EPN [34], 3D-ED-GAN [35]. However, voxelization will irreversibly lose a large number of geometric features and texture details. To address this, GR-NET [24] maps voxels back to point clouds, which preserves details to a certain extent, but voxel-based methods are limited by resolution, which requires a lot of memory and computational costs, and recent research has gradually abandoned them.

As the pioneer of point cloud learning, PointNet [11] solved the problem of point cloud disorder. A number of variations of point cloud-based learning algorithms have been derived such as PointNet++ [18], DGCNN [36], PointCNN [37], which has also led to the further development of point cloud-based completion methods. As a starting point for folding a 2D mesh into a 3D point cloud, FoldingNet [21] uses a 2D manifold to reconstruct a 3D point cloud by multiple folding. As the first algorithm dedicated to point cloud completion, PCN [22] generates a coarse point cloud through an encoding–decoding framework, and then maps to a 3D point cloud using a small-scale 2D grid. Thereafter, in order to restore more accurate geometrical shape information, the proposed TopNet [23] tree decoder does not include any topological structure on the point set, and more generally generates point clouds with arbitrary shapes. PF-Net [25] uses a multiscale encoder and pyramid decoder to extract and recover more information to a certain extent by multilevel generation of point clouds. Although the recent work on point cloud remediation such as MSN [38], SoftPoolNet [39], PMP [26], etc. has greatly improved the capability of point cloud completion, there are still a series of problems in the task of dense point clouds, such

as inaccurate structure remediation, insufficient smoothness and completeness of surface remediation, loss of details and generation of scattered points.

Transformer. Transformer originated from natural language processing and was gradually applied to computer vision [40]. Transformer is usually an encoder–decoder structure whose self-attention mechanism provides context information. Pioneer work on PCT [41] and Point Transformer [42] has already shown excellent results in point cloud classification and segmentation tasks. Transformer can effectively notice the local information and context information of the point cloud, which is particularly important in the point cloud completion task that requires more feature information.

3. Methods

3.1. Overview

The proposed MAPGNet aims to complete a sparse incomplete point cloud as a dense and complete point cloud in a coarse-to-fine manner. The network inputs a missing point cloud and outputs a complete and dense point cloud through an encoder–decoder frame structure. The framework of MAPGNet is shown in Figure 1. The framework is divided into three modules, namely the composite encoder feature extraction module, the skeleton point cloud generation module and the point cloud growth module. The details of each module and the loss function will be introduced separately below.

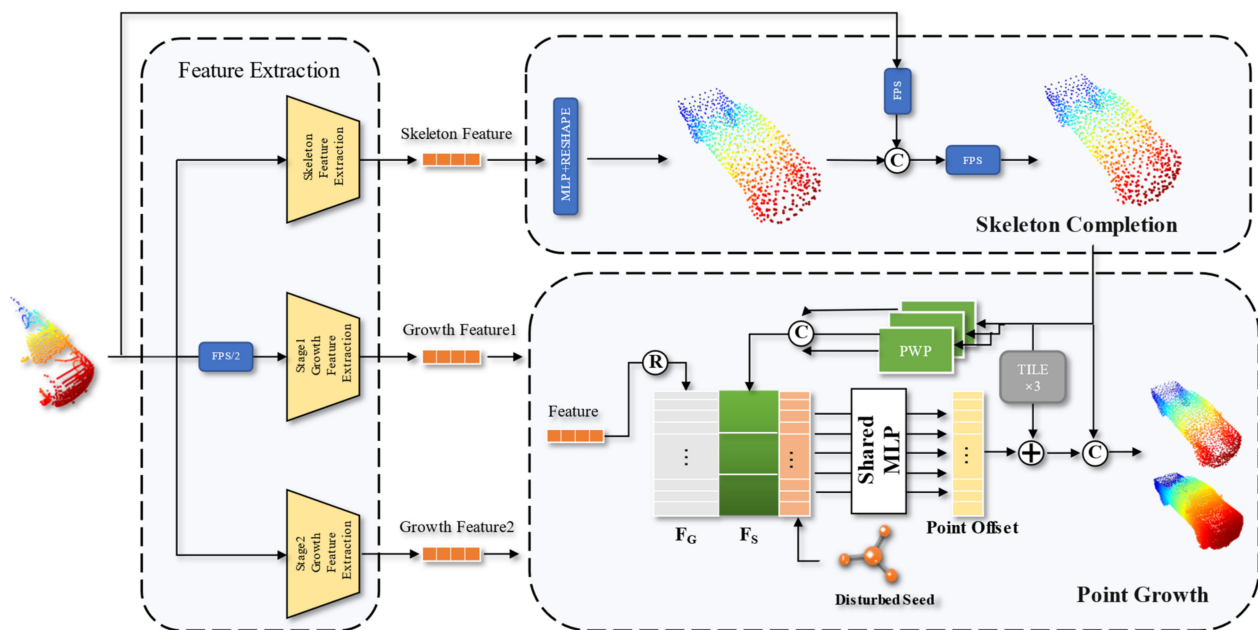


Figure 1. Overall framework structure of MAPGNet, composed of a composite encoder feature extraction module, a skeleton generation module and a point cloud growth module. FPS denotes the farthest point sampling. PWP is to extract the feature structure from the skeleton point cloud. @denotes the copy operation. © denotes concatenation operation. ⊕ denotes add operation. Tile ×3 denotes copying the skeleton point cloud three times to form an $N_C \times 3$ vector.

3.2. Composite Encoder Feature Extraction Module

The task of the feature extraction module is to extract the geometric information and local structure information of the input missing point cloud, and collect and summarize the information into the generated feature vector. Different from tasks such as classification and segmentation, point cloud completion tasks need to extract more features to recover the complex morphological structure and texture details of dense point clouds. Most of the completion tasks focus too much on the point cloud decoding process and ignore the importance of obtaining more features from the input missing point cloud. In this paper, we innovatively propose a composite encoder feature extraction module, which can extract

a large number of point cloud features in different phases to complete a more accurate and dense point cloud.

As shown in the Feature Extraction section of Figure 1, the module consists of three encoders, Skeleton Feature Extraction, Growth 1 Feature Extraction and Growth 2 Feature Extraction. Each encoder is responsible for encoding tasks in different point cloud generation stages, where the two encoders in Growth Feature Extraction are similarly used to extract the relational features of the skeleton point cloud to the growing point cloud, and the features of the local geometry and texture. The difference is that the Growth 2 Feature Extraction task extracts deeper details in a smaller area, so higher-resolution point clouds need to be input to extract more relevant local features.

The Skeleton Feature Extraction task is to extract the features of the initial skeleton shape point cloud. Since only the roughly complete shape of the point cloud and a small number of point clouds need to be recovered, the Skeleton Feature Extraction does not adopt a complex encoding structure. As shown in the Skeleton Feature Extraction section in Figure 2, the vector C is simply expanded and fused with the feature matrix in the PointNet, which is eventually extracted by max-pooling to the features that generate the skeleton point cloud.

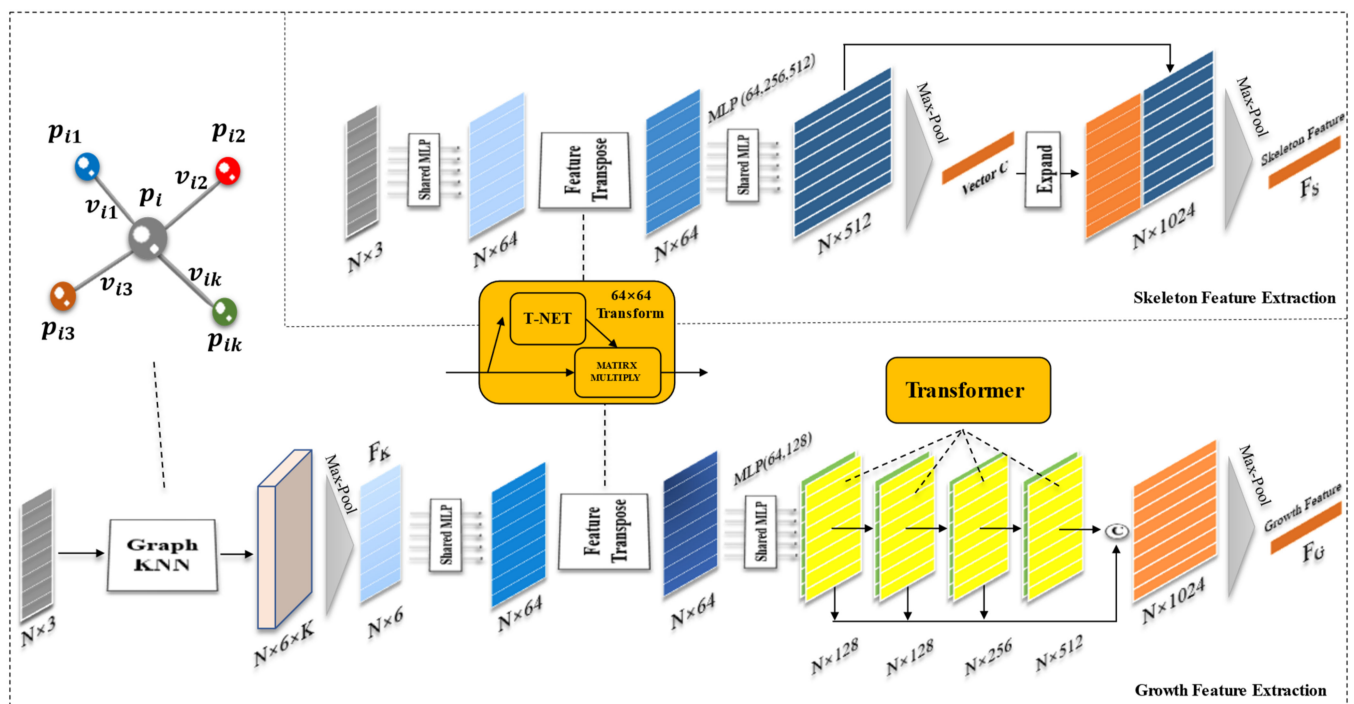


Figure 2. The structure of the composite encoder, including the skeleton point cloud feature extraction module and the dense point cloud feature extraction module. It converts the input incomplete point cloud to feature vectors for three different completion stages.

The next stage in the Growth Feature Extraction is to extract more complex geometric structure information at a small scale from the input point cloud, which is eventually assembled into a feature vector F_G . As shown in the Growth Feature Extraction section in Figure 2, in order to extract the local geometric structure information and context information of the point cloud more effectively, a local neighborhood map and Offset Transformer structure are introduced to enhance its ability to encode local point clouds. A local neighborhood graph is first constructed in the input point cloud, as shown in Figure 2 at Graph KNN, which consists of point p_i and directed edge v_{ij} . v_{ij} is the directed edge of the p_i relative to the neighboring point p_{ik} of p_i :

$$G = (P, V) \tag{1}$$

$$\text{Among : } \begin{cases} P = \{p_i | i = 1, 2, \dots, n\} \\ V = \{v_i = (v_{i1}, \dots, v_{ik}), | i = 1, 2, 3, \dots, n\} \end{cases} \quad (2)$$

$$\text{Among : } v_{ij} = p_{ij} - p_i | j = 1, 2, \dots, k, \quad (3)$$

where k is the number of neighboring points selected, and n is the number of point clouds. By expanding the feature P and splicing it with feature V in the neighborhood graph, a feature matrix containing local neighborhood information is obtained, and then the fusion feature F_K of the input point cloud is obtained after max-pooling.

Figure 3 shows the Offset Transformer structure introduced in the encoder. The purpose of introducing offset-attention in the feature expansion stage is to enhance the ability of feature relationship perception. While paying attention to the connection between the local structural features of the point cloud and the context, we also focus on the connection between the skeleton point clouds and the generation of denser point clouds. As shown in Figure 3, given a feature graph H , a new feature graph H' with an offset and attention mechanism is generated using the transformer with residual form.

$$H' = \text{Offset-Attention}(H) = V_H - \text{Attention}(Q_H, K_H, V_H). \quad (4)$$

$$\text{Among : } \begin{cases} \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dk}}\right)V, \\ Q_H, K_H, V_H = H \cdot (W_Q, W_K, W_V) \end{cases} \quad (5)$$

where Q, K, V , in the formula denote the query matrix, the key matrix and the value matrix computed from the input original characteristic linear transformation matrix W_Q, W_K, W_V . Attention (Q, K, V) is the attention mechanism transformer, $\text{softmax}(\cdot)$ is a normalized exponential function. The inner product of each row vector of the matrix Q and K is calculated. In order to prevent the inner product from being too large, a scaling factor $1/\sqrt{dk}$ is added. d_k denotes the dimensions of the K matrix. The offset-attention further biases the feature map on this basis, focusing more on the structural and spatial relationships between the skeleton point cloud and the generated point cloud, and mitigating the influence of noise points and unimportant similar features on the overall features, so as to recover a more accurate point cloud model.

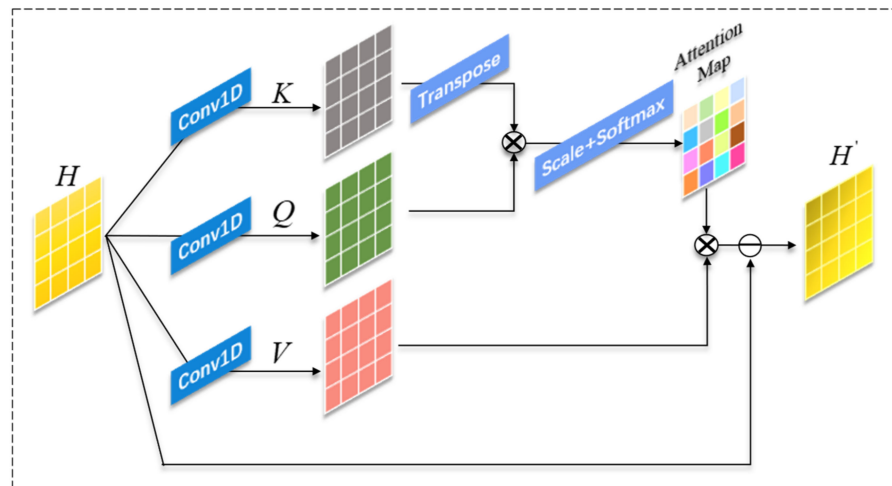


Figure 3. Details of the Offset Transformer structure. \oplus denotes add operation. \ominus denotes subtraction operation.

3.3. Skeleton Point Cloud Generation Module

The goal of the skeleton point cloud generation module is to generate a sparse and complete point cloud. As shown in Skeleton Completion of Figure 1, the module uses F_C extracted by Skeleton Feature Extraction Encoder to generate a complete and accurate geometric shape of the point cloud. The PCN network [22] confirms that the fully connected

decoder is better at predicting the global geometry. The skeleton point cloud generation module first adopts a fully connected network and reshapes it into a point cloud P_{C0} of $N_C \times 3$. In order to make full use of the input information of the missing point cloud and effectively add only accurate point clouds to the skeleton point cloud, we merge the P_{C0} with the missing point cloud, and then sample the merged point cloud to the new skeleton point cloud P_C of $N_C \times 3$ through farthest point sampling (FPS). P_C is the skeleton point cloud to be extended on this basis.

3.4. Point Cloud Growth Module

The task of the point cloud growth module is to grow new local point clouds based on the skeleton point cloud, so as to expand the number of point clouds and refine the point cloud details. Networks such as FoldingNet [21] and PCN [22] expand the number of point clouds based on folding, which expands the number of point clouds by adding a fixed 2D mesh that is slightly perturbed to deviate from the original point cloud position, but the dense point clouds they generate are not uniform and smooth, and even multiple mesh facets overlap each other at locations with complex point cloud structures. In order to solve the problems based on the folding approach, a new point cloud growth method is proposed in this paper. The point cloud adaptively grows new point clouds within the spherical neighborhood of each skeleton point. The method as in Figure 4 increases the number of point clouds by the two-step point cloud growth module as a way to achieve accurate recovery of dense target point clouds.

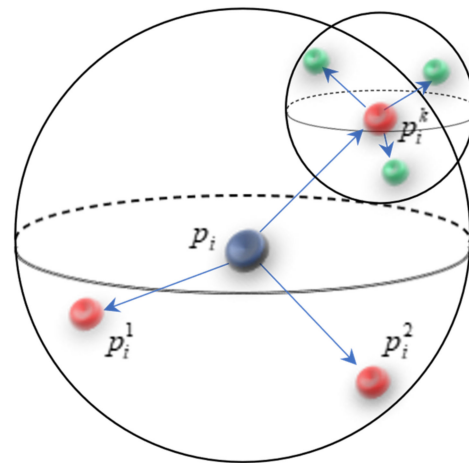


Figure 4. Two-step growth illustration in point cloud neighborhood.

Specifically, the point cloud growth module is based on the skeleton point cloud P_C , using each independent skeleton point as the starting point p_i , and adaptively growing the required points p_i^k (k represents the point cloud expansion multiplier) around the limited sphere neighborhood of the starting point p_i . Different from the folding-based method, the point cloud growth module fully considers the geometric features of the generated skeleton point cloud. Thus, as in the Point Growth section of Figure 1, we use the PointNet structure without pooling (PWP) to upgrade the feature of the skeleton point cloud to obtain the feature map F_S of the skeleton point cloud. Higher-dimensional features help predict the relationship between the skeleton point cloud and its corresponding growth point cloud. The feature vector extracted by the Growth 1/2 Feature Extraction Encoder is repeated k times (F_D) and then concatenated with the feature map F_S of the skeleton point cloud. Meanwhile, the point cloud growth module also introduces different direction perturbation coordinates x_i^k , which ensure that the point cloud grows in different directions, and solves the problem of point overlap. Then, a shared-MLP and tanh are used to generate offset vectors:

$$\Delta p_i^k = l \times \tanh(\text{MLP}(\text{concat}(F_D, F_S, x_i^k))), \quad (6)$$

where $\tanh(\cdot)$ denotes the hyper-tangent activation, $\text{MLP}(\cdot)$ denotes multilayer perceptron, $\text{concat}(\cdot)$ denotes the concatenation operation, l denotes the radius of the sphere's neighborhood.

The coordinates of the final growing point p_i^k are:

$$p_i^k = p_i + \Delta p_i^k. \quad (7)$$

On the basis of the skeleton point cloud, the point cloud is grown adaptively in k different directions in a predefined spherical neighborhood each time. Within the surface, fitting to the surface, in the edge area, the point cloud grows in the direction of the inside of the object. As shown in Figure 4, the point cloud growth module designed in this paper is divided into two stages, with the neighborhood radius of the two growth stages being reduced step by step to grow local point clouds of different resolutions. The skeleton point cloud utilized in the second stage is the point cloud generated in the previous stage. The two-stage point cloud growth does not restrict the specific direction and distance of the point cloud growth, and grows the local detail point cloud in a more free manner based on the already generated point cloud.

3.5. Training Loss

Point cloud completion tasks usually employ chamfer distance (CD) and Earth mover distance (EMD) as loss functions to calculate the difference between point clouds [22,24,26]. Since the design is a multilevel generation module, we sample (FPS) the ground truth point cloud into point cloud $\{G_C, G_1, G_2\}$ with the same resolution as the multilevel generated point cloud $\{P_C, P_1, P_2\}$. Compared with the CD, the EMD is more sensitive to the geometric integrity of the point cloud, but it also has limitations such as a large amount of calculation and the same sized point cloud. Therefore, in the initially generated skeleton point cloud P_C , the EMD is used as the loss function, and the EMD is defined as:

$$\mathcal{L}_{\text{EMD}}(P_C, G_C) = \min_{\phi: P_C \rightarrow G_C} \frac{1}{|P_C|} \sum_{x \in P_C} \|x - \phi(x)\|, \quad (8)$$

where ϕ is a bijection that minimizes the distance between the corresponding point of the generated point cloud P_C and the ground truth point cloud G_C . The EMD is the minimum cost of converting two point clouds, but the cost of finding the optimal ϕ is too expensive, so iterative $(1 + \varepsilon)$ is used to approximate [22].

In the point cloud growth phase, we adopt a symmetric version of the chamfer distance (CD), defined as:

$$\mathcal{L}_{\text{CD}}(P_i, G_i) = \frac{1}{P_i} \sum_{x \in P_i} \min_{y \in G_i} \|x - y\| + \frac{1}{G_i} \sum_{y \in G_i} \min_{x \in P_i} \|y - x\|, \quad (9)$$

where CD is the average shortest point distance between the generated point cloud and the ground truth point cloud. The first half ensures the minimum distance between the generated point cloud and the ground truth point cloud, and the second half ensures the coverage of the ground truth point cloud in the generated point cloud.

Therefore, we define the total training loss as:

$$\mathcal{L} = \mathcal{L}_{\text{EMD}}(P_C, G_C) + \sum_{i=1}^2 \mathcal{L}_{\text{CD}}(P_i, G_i) \quad (10)$$

4. Experiments

In this section, in order to verify the effectiveness of our algorithm, we conduct analysis experiments on two standard point cloud completion datasets, PCN and Completion3D.

4.1. Dataset

PCN dataset. The PCN dataset is derived from the article on PCN [22], which is the most commonly used benchmark for dense point cloud completion. There are eight categories and 30,974 models in PCN. Each model generates point clouds of eight missing parts from eight randomly distributed perspectives. The input missing point cloud contains 2048 points, and the corresponding complete point cloud contains 16,384 points. For uniform evaluation, we adopt the same L1-CD as the previous completion method.

Completion3D dataset. The Completion3D dataset is derived from the article on TopNet [23] and is a subset of the ShapeNet dataset. In Completion3D, part of the point cloud is generated by projecting the 2.5D depth image in the reverse direction into 3D, and the input point cloud and the ground truth point cloud are obtained by random sampling, and there are 2048. To perform a uniform evaluation on this dataset, we used the same L2-CD as in the previous completion methods.

4.2. Metrics

In order to compare with the previous completion algorithm at the same scale, we choose the same measurement scale L1/L2 chamfer distance as the previous study as the evaluation criterion. Assuming that the predicted point cloud set is X , the ground truth is Y and the number of points in the point cloud is n , the L1-CD formula is:

$$\mathcal{L}_{CD-L1}(X, Y) = \frac{1}{nX} \sum_{x \in X} \min_{y \in Y} \|x - y\| + \frac{1}{nY} \sum_{y \in Y} \min_{x \in X} \|y - x\|. \quad (11)$$

The L2-CD version replaces Equation (11) with the L2 norm.

$$\text{F-Score}(d) = \frac{2P(d)R(d)}{P(d) + R(d)} \quad (12)$$

However, it is pointed out in [43] that the chamfering distance is sometimes insufficient to describe the difference between the two point clouds, so we introduce the F-score as a supplementary measure.

$P(d)$, $R(d)$ denote precision and recall under the threshold value d :

$$P(d) = \frac{1}{nX} \sum_{x \in X} \left[\min_{y \in Y} \|x - y\| < d \right] \quad (13)$$

$$R(d) = \frac{1}{nY} \sum_{y \in Y} \left[\min_{x \in X} \|x - y\| < d \right] \quad (14)$$

4.3. Implementation Details

The proposed framework is implemented in Pytorch in Python and is trained on an NVIDIA 3080TI GPU. The Adam optimizer is adopted for the model with 300 epochs, and the initialization learning rate is 0.0001. The learning rate is decayed by 0.8 every 50 epochs. The batch size is set to 16. On the PCN dataset, we generate 1024 skeleton points, and then generate 16,384 points in two steps. On the Completion3D dataset, 512 skeleton points are generated, and 2048 points are generated in two steps.

4.4. Completion Results on PCN

On the PCN dataset, our results are compared with those of other advanced point cloud completion methods, the quantitative comparison results are shown in Tables 1 and 2 and Figure 5 shows the visualization comparison results.

Table 1. Point completion results on PCN dataset in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . The best results are highlighted in bold (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
3D-EPN [34]	20.15	13.16	21.8	20.31	18.81	25.75	21.09	21.72	18.54
POINTNET++ [18]	14	10.3	14.74	12.19	15.78	17.62	16.18	11.68	13.52
FOLDINGNET [21]	14.31	9.49	15.8	12.61	15.55	16.41	15.97	13.65	14.99
TOPNET [23]	12.15	7.61	13.31	10.9	13.82	14.44	14.78	11.22	11.12
ATLASNET [44]	10.85	6.37	11.94	10.1	12.06	12.37	12.99	10.33	10.61
PCN [22]	9.64	5.5	22.7	10.63	10.99	11	11.34	11.68	8.59
SOFTPOOLNET [39]	9.205	6.93	10.91	9.78	9.56	8.59	11.22	8.51	8.14
MSN [38]	9.97	5.6	11.96	10.78	10.62	10.71	11.9	8.7	9.49
GRNET [24]	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
PMP [26]	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
OURS	8.59	4.85	10.44	8.32	9.95	7.56	11.15	8.31	8.18

Table 2. Point completion results on PCN dataset in terms of F-score computed on 16,384 points. The best results are highlighted in bold (higher is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
ATLASNET [44]	0.616	0.845	0.552	0.630	0.552	0.565	0.500	0.660	0.624
PCN [22]	0.695	0.881	0.651	0.725	0.625	0.638	0.581	0.765	0.697
FOLDINGNET [21]	0.322	0.642	0.237	0.382	0.236	0.219	0.197	0.361	0.299
TOPNET [23]	0.503	0.771	0.404	0.544	0.413	0.408	0.350	0.572	0.560
MSN [38]	0.705	0.885	0.644	0.665	0.657	0.699	0.604	0.782	0.708
GRNET [24]	0.708	0.843	0.618	0.682	0.673	0.761	0.605	0.751	0.750
OURS	0.729	0.913	0.650	0.749	0.680	0.749	0.612	0.788	0.754

Quantitative Comparison. The comparison between MAPGNet and other advanced methods for point cloud completion is shown in Tables 1 and 2. Ours is superior to other methods on the average L1-CD of all categories in the PCN dataset, improving 1.6% over the second best PMP algorithm, which proves that ours has certain generalization ability in completing different shape categories and has the best performance index. In the categories of plane, car and table, ours achieved the best results in the measurement of L1-CD. MAPGNet achieved better results in four categories compared to PMP. Overall, ours has better performance. Compared with GRNet, GRNet’s cabinet and couch are better than ours in indicators, because GRNet is based on 3DCNN, which is more suitable for completing some planar and cubic objects. PCN, SoftpoolNet, MSN and MAPGNet are all coarse-to-fine point cloud completion methods, in which we also show better performance indicators. Meanwhile, in the supplementary measure of F-score in Table 2, ours improved 3% over GRNet and achieved the best of the six categories and average results, which shows that our algorithm performs better. Therefore, the best results are achieved in the task of completing dense point clouds of PCN datasets.

Visual Comparison. To further evaluate MAPGNet’s ability to complete missing point clouds, in Figure 5, an intuitive visualization compares MAPGNet with other advanced algorithms, and our method achieves better visual effects. The dense point cloud completed by MAPGNet has more accurate local structure, smoother surface and less scattered points. At the same time, the edge of the predicted point cloud is sharper and can accurately describe the range of the edge. The example we selected is the more complex missing point cloud. Our method can accurately predict the complex details such as the blade of the aircraft, the armrest of the chair and the leg of the table while retaining the missing point cloud. Meanwhile, the generated point cloud distribution is very uniform and dense. However, PCN and GRNet have been unable to predict the part of the local missing detail point cloud. Essentially, PMP increases the number of points by multiple moving points,

which will have many overlapping points when generating dense point clouds, and the distribution of point clouds is very sparse, which can also be confirmed in the visual comparison in Figure 5.

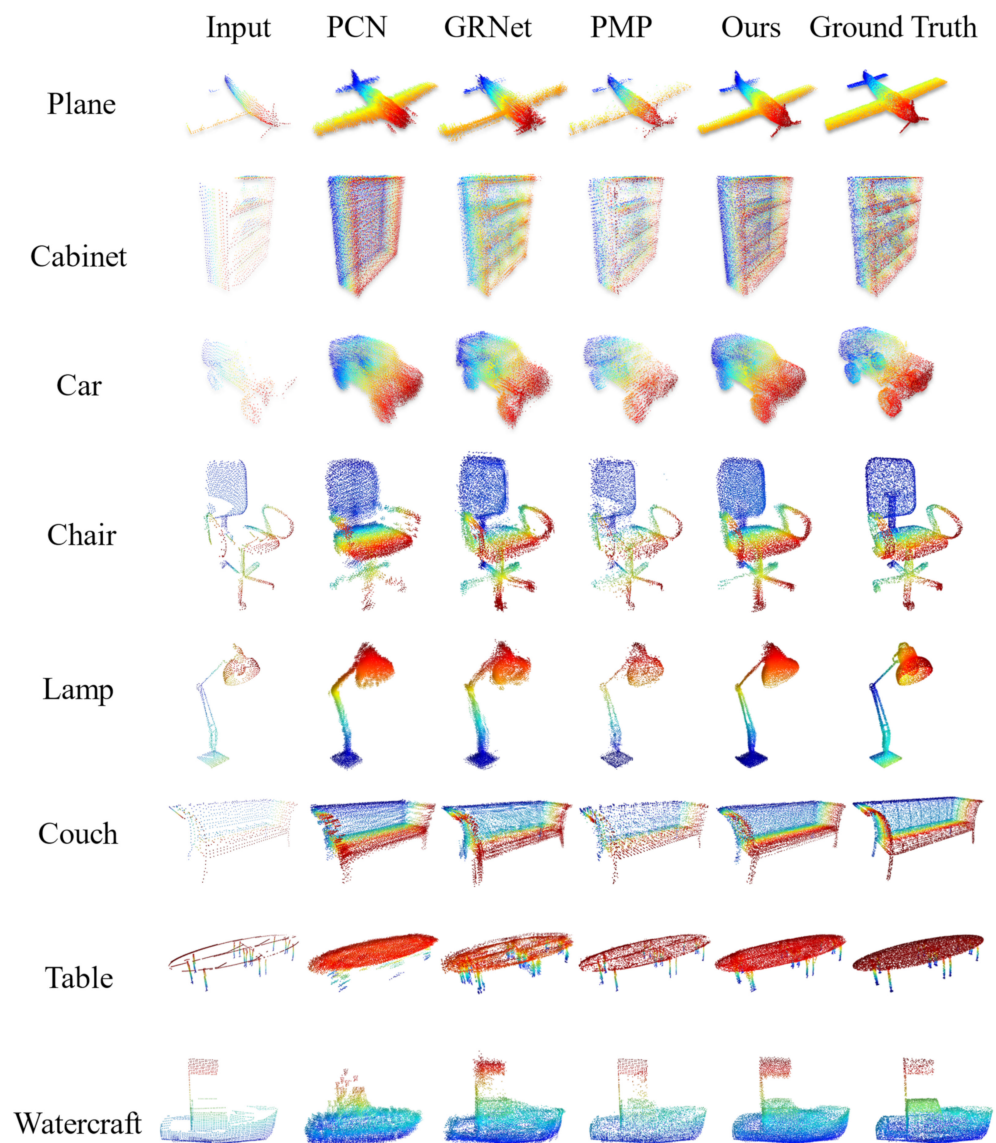


Figure 5. Comparison of visualization results on PCN dataset. Compared with other advanced algorithms, MAPGNet (Ours) has better visualization completion effect and can produce more accurate geometric structures and smoother surfaces. Ground Truth is the real complete point cloud corresponding to the Input.

4.5. Completion Results on Completion3D

With the Completion3D dataset, we visually compare our completion results with other advanced methods, and the results of the comparison are shown in Table 3 and specifically in Figure 6.

Quantitative Comparison and Visual Comparison. The comparison of MAPGNet with other advanced algorithms on the Completion3D dataset is shown in Table 3. Our method is mainly aimed at the completion of dense point clouds. Since the point cloud volume of Completion3D is 2048, it cannot fully demonstrate the ability of our algorithm to predict surfaces. However, it still shows good performance in the overall average value and L2-CD of airplane, cabinet, car, couch and watercraft. In Figure 6, the input point cloud with a large missing area is selected. We can see that when PMP (the second best algorithm

in Table 3) cannot accurately predict the geometric structure of the object, our algorithm can still accurately predict it. This is because the PMP method is based on the complementation of moving points. In the case of wide-range missing, the predicted point of original point cloud movement cannot reach the missing position accurately, which is the reason why PMP failed to forecast under the complex large-range missing point cloud. Our method has a natural advantage over PMP because it can cover the whole missing part and make up the geometric details when predicting large-scale missing. Therefore, our method also achieves the best result on Completion3D.

Table 3. Point completion results on Completion3D dataset in terms of per-point chamfer distance (CD) with L2 norm computed on 2048 points and multiplied by 10^4 . The best results are highlighted in bold (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
FOLDINGNET [21]	19.07	12.83	23.01	14.88	25.69	21.79	21.31	20.71	11.51
PCN [22]	18.22	9.79	22.7	12.43	25.14	22.72	20.26	20.27	11.73
POINTSETV [45]	18.18	6.88	21.18	15.78	22.54	18.78	28.39	19.96	11.16
ATLASNET [44]	17.77	10.36	23.4	13.4	24.16	20.24	20.82	17.52	11.62
SOFTPOOLNET [39]	16.15	5.81	24.53	11.35	23.63	18.54	20.34	16.89	7.14
TOPNET [23]	14.25	7.32	18.77	12.88	19.82	14.6	16.29	14.89	8.82
SA-NET [46]	11.22	5.27	14.45	7.78	13.67	13.53	14.22	11.75	8.84
GRNET [24]	10.64	6.13	16.9	8.27	12.23	10.22	14.93	10.08	5.86
PMP [26]	9.23	3.99	14.7	8.55	10.21	9.27	12.43	8.51	5.77
OURS	8.87	2.92	13.53	6.01	11.05	10.76	9.15	11.87	5.71

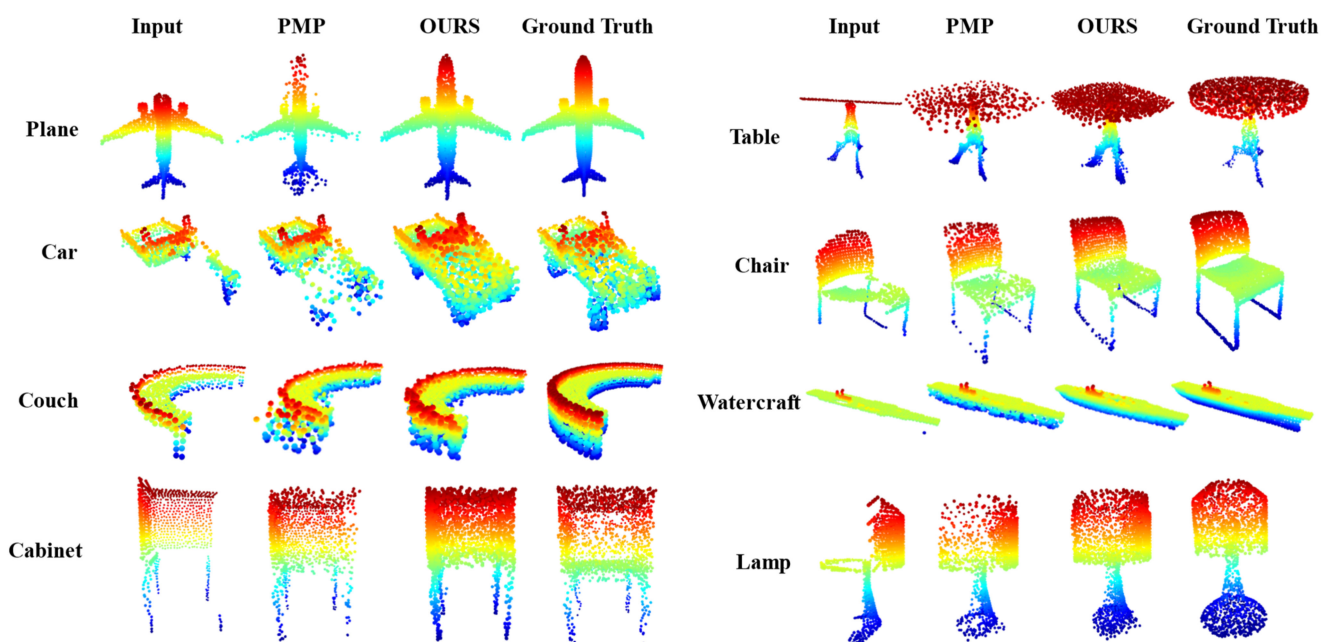


Figure 6. Visual comparison with PMP (the second best algorithm) on the Completion3D dataset. The visualizations completed by MAPGNet (OURS) have a more accurate geometry. Ground Truth is the real complete point cloud corresponding to the Input.

5. Model Analysis

In order to verify the impact of each part of our algorithm on the overall network system, we analyze the modules of the different parts proposed. Aircraft, car, couch and watercraft were selected for verification experiments in the PCN dataset. By default, we only remove or change the structure of the part that needs to be analyzed, while the network structure of other parts remains unchanged.

5.1. Analysis of Composite Encoder Module

We analyze the effectiveness of the composite encoder by changing the composite encoding (CE) to a single encoder and changing the type of encoder. The experiments on encoder modules are divided in the following ways:

- (1) Change encoder to PointNet [11].
- (2) Change the encoder to PointNet++ [18].
- (3) Change encoder to DGCNN [36].

By comparing the impact of different encoders on the final results, it can be seen from Table 4 that the final L1-CD metric can be improved significantly with the composite encoder. Utilizing the same features at different stages can lose information about the point cloud details. Compared with other encoders, the composite encoder uses different features at different stages, which enables the encoder to focus more on that part of the features at that stage, thus recovering more accurate details of the point cloud.

Table 4. Analysis and comparison results of different encoders in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . The best results are highlighted in bold (lower is better).

Evaluate	Avg.	Airplane	Car	Couch	Watercraft
PointNet	9.125	5.72	9.32	12.15	9.23
PointNet++	8.53	5.50	8.97	11.43	8.22
DGCNN	8.57	5.53	8.78	11.46	8.51
CE	8.13	4.85	8.32	11.15	8.18

5.2. Analysis of Limited Sphere Neighborhood Radius

The size of the radius of the ball neighborhood of the point cloud growth module has an impact on the final generated results. In the process of growing the dense point cloud in two stages after generating the skeleton point cloud, the radius of the point cloud grown in the first stage is r_1 , and the radius of the point cloud grown in the second stage is r_2 . The following experiments are conducted with spherical neighborhoods of different radii.

According to Table 5 and Figure 7, when the sphere neighborhood radius is set too large, some of the points growing near the skeleton point will deviate from other skeleton points, which will eventually result in a non-smooth surface, and even a slight shadow and scattering of points. When the radius of the sphere neighborhood is set too small, the points growing near the skeleton points will appear aggregated, and there are irregular holes on the generated surface. In serious cases, there will be problems such as sparse and uneven distribution of point clouds caused by point cloud overlap. Therefore, considering that the model needs to be suitable for multiple types of problems, we finally select the radius of the two-level spherical neighborhood with the best index as [0.2, 0.1].

Table 5. Comparison of results for different radii in the sphere neighborhood in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . The best results are highlighted in bold (lower is better).

Radius	Avg.	Airplane	Car	Couch	Watercraft
[0.5, 0.25]	8.46	5.05	8.9	11.51	8.39
[0.4, 0.2]	8.26	5.07	8.7	11.02	8.27
[0.2, 0.1]	8.13	4.85	8.32	11.15	8.18
[0.1, 0.05]	8.29	5.11	8.27	11.32	8.46
[0.02, 0.01]	8.61	5.32	8.86	11.63	8.59

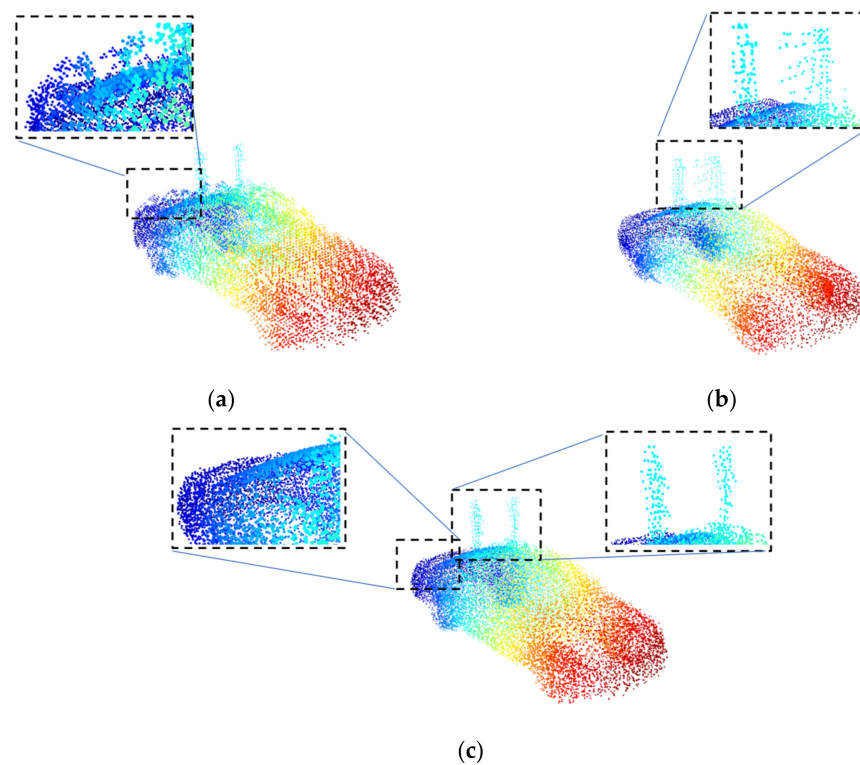


Figure 7. Visualization results for different neighborhood radii. The radius of (a) is too large, (b) too small, (c) applicable.

5.3. Analysis of Offset Transformer Structure

Four groups of Offset Transformer are added to the network structure. In order to evaluate the impact of Offset Transformer, we designed the following three experiments:

- (1) Remove Offset Transformer.
- (2) Replace Offset Transformer with channel-attention mechanism SE-Net [47].
- (3) Replace Offset Transformer with normal Transformer.

In Table 6, by removing the attention mechanism module, it is found that Offset Transformer can improve the performance indicators of the model, which should be related to the fact that Transformer can notice the local information and contact context of the point cloud. By comparing the results of different attention mechanism modules in the above table, we can also find that Offset Transformer has better results than the other two attention mechanisms, which also confirms the effectiveness of Offset Transformer. The Offset Transformer module in the Growth Feature Extraction can focus on more useful local features and associate them with the features of the skeleton point cloud, so that it can extract more useful detail information in the encoding process.

Table 6. The effect of Offset Transformer on network performance in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . The best results are highlighted in bold (lower is better).

Evaluate	Avg.	Airplane	Car	Couch	Watercraft
w/o Attention	8.35	4.94	8.55	11.53	8.38
SE	8.26	5.01	8.33	11.46	8.22
Normal Transformer	8.16	4.92	8.37	11.32	8.06
Offset Transformer	8.13	4.85	8.32	11.15	8.18

5.4. Analysis of PWP Decoding Structure

The point cloud generation module needs to expand the number of point clouds exponentially, where the features of the skeleton point cloud need to be spliced and fused with the features in dense feature extraction (DFE). Each point of the skeleton point cloud is up-dimensioned with a PointNet structure without pooling and then stitched. We replace the structure of PWP with the three-dimensional coordinates of the skeleton point cloud and the local folding operation in PCN. The following three experiments were designed separately.

- (1) Splicing directly with the skeleton point coordinates and DFE features.
- (2) Splicing DFE features using folding in PCN.
- (3) No disturbance vector.

In Table 7, it is obvious that up-dimensioning the skeleton point cloud by using the PWP structure to up-dimension the features can substantially improve the final performance metrics. Compared to replacing it with a folding structure in PCN, our method improves 7.8% in L1-CD and 13.9% in metrics compared to the direct stitching with coordinates of points. The disturbance vector added by the growth module can slightly improve the final results, because different disturbance vectors can avoid excessive overlapping of points grown in the neighborhood of the sphere. The point cloud growth module allows point clouds to be grown freely in the spherical neighborhood, rather than folding into three-dimensional point coordinates as two-dimensional grid points. In Figure 8, it is clear that our method is more flexible and versatile compared to the folding decoding structure, sharper at the edges of the point cloud model and smoother and more uniform where the point clouds are distributed over large surfaces.

Table 7. Analysis and comparison of PWP in the point cloud growth module in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . The best results are highlighted in bold (lower is better).

Evaluate	Avg.	Airplane	Car	Couch	Watercraft
Point cloud coordinates	9.26	5.76	10.65	11.88	8.74
PCN-FOLDING	8.82	5.25	10.37	11.24	8.42
w/o disturb	8.21	4.96	8.43	11.25	8.19
PWP	8.13	4.85	8.32	11.15	8.18

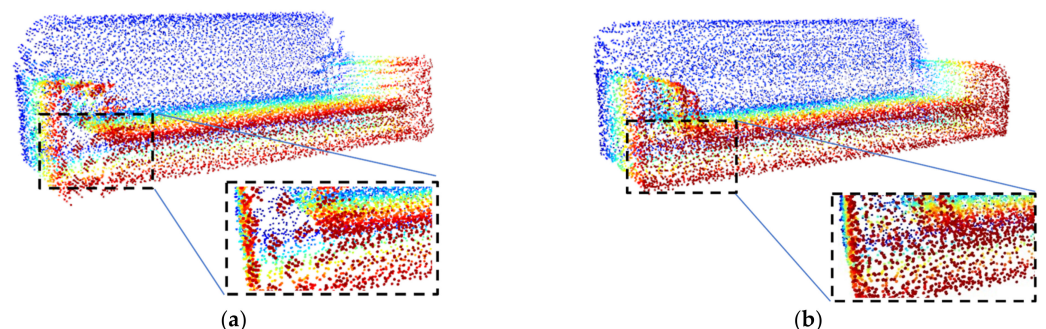


Figure 8. Visual comparison of our PWP structure and folding structure. (a) Folding, (b) PWP.

5.5. Ablation Experiments

The improved performance of MAPGNet is mainly attributed to three key components: the design of the composite encoder (CE), the PWP structure in the growth module and the Offset Transformer. By comparing and analyzing the data in Tables 4, 6 and 7, we can prove the effectiveness of each component, and the indicator is the specific improvement percentage.

From Table 8, we can see that the PWP can be improved by 8.49% compared with the traditional folding structure, which shows that the decoding is very important for the

characterization of local details. The composite encoder can extract more missing features from the input point cloud, which is 4.8% higher than that of PointNet++. As an attention mechanism structure, Offset Transformer can also improve the performance by 2.71%. All in all, these three modules improve the performance of the system to a certain extent and play an indispensable role.

Table 8. Ablation experiments on dataset PCN, the results in terms of per-point chamfer distance (CD) with L1 norm computed on 16,384 points and multiplied by 10^3 . Enhance percent is the influence of the module on the overall promotion.

	MAPGNet w/o Offset Transformer	MAPGNet w/o CE + POINTNET++	MAPGNet w/o PWP + FOLDING	MAPGNet
Avg.	8.35	8.53	8.82	8.13
Enhance Percent	2.71%	4.82%	8.49%	/

6. Discussion

We visualized all completed objects in the test set, and we found that the MAPGNet did not predict accurately in both cases.

The input point cloud is too small in Figure 9, resulting in too little 3D feature information. Multiple similar objects may contain a common part. When the input point cloud is the common part, the completed result will be confused, and it may be completed as other similar objects. Currently, almost all algorithms cannot solve this problem.

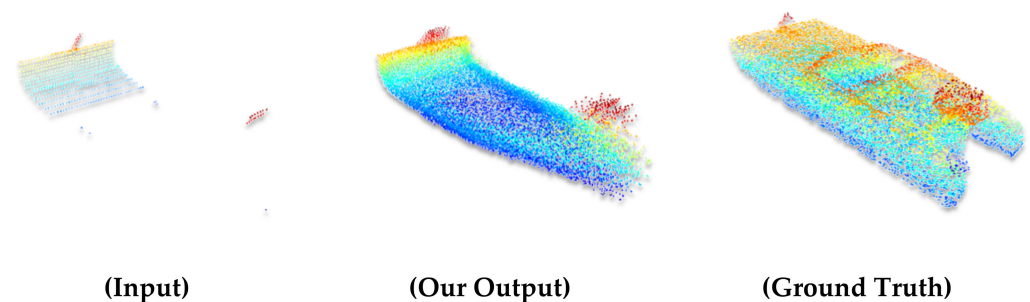


Figure 9. Failure example of the part of the input point cloud being too small.

In Figure 10, the independent structures in the same object are not connected, and the independent parts may be very close to each other. Our method sometimes connects adjacent independent structures together incorrectly, so the effect of complementing objects with independent structures is not good. In this case, there may be too few objects with independent structures in the training set, and the prior knowledge cannot be fully learned.

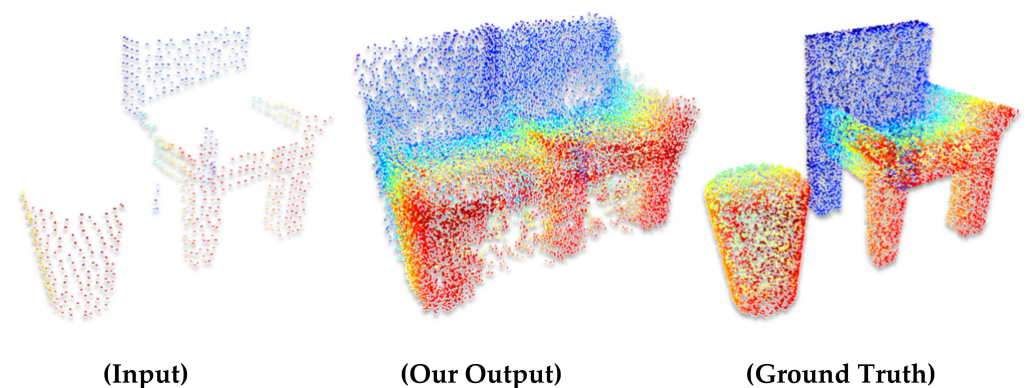


Figure 10. Failure example of objects containing independent structures.

7. Conclusions

In this paper, we propose a novel adaptive point-growth network MAPGNet for dense point cloud completion, which extracts the features of missing point clouds sufficiently by designing multiple encoders, and then generates dense and complete point clouds in spherical neighborhoods adaptively through two-stage point cloud growth modules. It solves the problems of inaccurate completion of complex structures, insufficient smoothness and completeness of surface completion, loss of details and scattered points in dense point cloud completion tasks. We analyze several existing advanced point cloud completion algorithms. The experimental results prove that our method achieves better results in different point cloud completion tasks, and shows that our method has excellent reconstruction and completion capabilities for different kinds of point cloud objects.

Author Contributions: Conceptualization, R.H.; methodology, R.H.; software, R.H.; validation, R.H., X.H. and K.Z.; formal analysis, R.H. and J.W.; investigation, R.H.; resources, X.H. and J.H.; data curation, R.H.; writing—original draft preparation, R.H. and Z.W.; writing—review and editing, Z.W.; visualization, R.H.; supervision, Z.W. and J.H.; project administration, R.H.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Changchun Scientific and Technological Development Program (No. 19SS008).

Data Availability Statement: PCN dataset (https://drive.google.com/file/d/1OvvRyx02-C_DkzYiJ5stpin0mnXydHQ7/view?usp=sharing, accessed on 1 February 2022). Completion3D dataset (<https://github.com/lynetcha/completion3d>, accessed on 1 February 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Q.; Kim, M.-K. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Adv. Eng. Inform.* **2019**, *39*, 306–319.
2. Bisheng, Y.; Fuxun, L.; Ronggang, H. Progress, challenges and perspectives of 3D LiDAR point cloud processing. *Acta Geod. Et Cartogr. Sin.* **2017**, *46*, 1509.
3. Horaud, R.; Hansard, M.; Evangelidis, G.; Ménier, C. An overview of depth cameras and range scanners based on time-of-flight technologies. *Mach. Vis. Appl.* **2016**, *27*, 1005–1020. [[CrossRef](#)]
4. Teppati Losè, L.; Spreafico, A.; Chiabrande, F.; Giulio Tonolo, F. Apple LiDAR Sensor for 3D Surveying: Tests and Results in the Cultural Heritage Domain. *Remote Sens.* **2022**, *14*, 4157. [[CrossRef](#)]
5. Van Oosterom, P.; Martinez-Rubi, O.; Ivanova, M.; Horhammer, M.; Geringer, D.; Ravada, S.; Tijssen, T.; Kodde, M.; Gonçalves, R. Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Comput. Graph.* **2015**, *49*, 92–125. [[CrossRef](#)]
6. Pang, G.; Qiu, R.; Huang, J.; You, S.; Neumann, U. Automatic 3D industrial point cloud modeling and recognition. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015.
7. Pérez, L.; Rodríguez, Í.; Rodríguez, N.; Usamentiaga, R.; García, D.F. Robot guidance using machine vision techniques in industrial environments: A comparative review. *Sensors* **2016**, *16*, 335. [[CrossRef](#)]
8. Kim, P.; Chen, J.; Cho, Y.K. SLAM-driven robotic mapping and registration of 3D point clouds. *Autom. Constr.* **2018**, *89*, 38–48. [[CrossRef](#)]
9. Pi, D.; Liu, J.; Wang, Y. Review of computer-generated hologram algorithms for color dynamic holographic three-dimensional display. *Light Sci. Appl.* **2022**, *11*, 1–17. [[CrossRef](#)]
10. Iglesias, L.; De Santos-Berbel, C.; Pascual, V.; Castro, M. Using Small Unmanned Aerial Vehicle in 3D Modeling of Highways with Tree-Covered Roadsides to Estimate Sight Distance. *Remote Sens.* **2019**, *11*, 2625. [[CrossRef](#)]
11. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
12. Uy, M.A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
13. Li, J.; Chen, B.; Lee, G.H. So-net: Self-organizing network for point cloud analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
14. Zhu, K.; He, X.; Gao, Y.; Hao, R.; Wei, Z.; Long, B.; Mu, Z.; Wang, J. Invalid point removal method based on error energy function in fringe projection profilometry. *Results Phys.* **2022**, *41*, 105904. [[CrossRef](#)]
15. Song, W.; Li, D.; Sun, S.; Zhang, L.; Xin, Y.; Sung, Y.; Choi, R. 2D&3DHNNet for 3D Object Classification in LiDAR Point Cloud. *Remote Sens.* **2022**, *14*, 3146.

16. Singer, N.; Asari, V.K. View-Agnostic Point Cloud Generation for Occlusion Reduction in Aerial Lidar. *Remote Sens.* **2022**, *14*, 2955. [[CrossRef](#)]
17. Liu, G.; Wei, S.; Zhong, S.; Huang, S.; Zhong, R. Reconstruction of Indoor Navigation Elements for Point Cloud of Buildings with Occlusions and Openings by Wall Segment Restoration from Indoor Context Labeling. *Remote Sens.* **2022**, *14*, 4275. [[CrossRef](#)]
18. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.
19. Ni, P.; Zhang, W.; Zhu, X.; Cao, Q. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
20. Chen, Y.; Liu, G.; Xu, Y.; Pan, P.; Xing, Y. PointNet++ network architecture with individual point level and global features on centroid for ALS point cloud classification. *Remote Sens.* **2021**, *13*, 472. [[CrossRef](#)]
21. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
22. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. Pcn: Point completion network. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018.
23. Tchapmi, L.P.; Kosaraju, V.; Rezaatofighi, H.; Reid, I.; Savarese, S. Topnet: Structural point cloud decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
24. Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; Sun, W. Grnet: Gridding residual network for dense point cloud completion. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020.
25. Zhang, J.; Shao, J.; Chen, J.; Yang, D.; Liang, B.; Liang, R. PFNet: An unsupervised deep network for polarization image fusion. *Opt. Lett.* **2020**, *45*, 1507–1510.
26. Wen, X.; Xiang, P.; Han, Z.; Cao, Y.-P.; Wan, P.; Zheng, W.; Liu, Y.-S. Pmp-net: Point cloud completion by learning multi-step point moving paths. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
27. Carr, J.C.; Beatson, R.K.; Cherrie, J.B.; Mitchell, T.J.; Fright, W.R.; McCallum, B.C.; Evans, T.R. Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001.
28. Li, Y.; Dai, A.; Guibas, L.; Nießner, M. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2015.
29. Pauly, M.; Mitra, N.J.; Wallner, J.; Pottmann, H.; Guibas, L.J. Discovering structural regularity in 3D geometry. In *ACM SIGGRAPH 2008 Papers*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 1–11.
30. Gupta, S.; Arbeláez, P.; Girshick, R.; Malik, J. Aligning 3D models to RGB-D images of cluttered scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
31. Su, H.; Maji, S.; Kalogerakis, E. Learned-Miller Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
32. Pang, G.; Neumann, U. 3D point cloud object detection with multi-view convolutional neural network. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016.
33. Yang, B.; Wen, H.; Wang, S.; Clark, R.; Markham, A.; Trigoni, N. 3D object reconstruction from a single depth view with adversarial learning. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017.
34. Dai, A.; Qi, C.R.; Nießner, M. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
35. Wang, W.; Huang, Q.; You, S.; Yang, C.; Neumann, U. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
36. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *Acm Trans. Graph.* **2019**, *38*, 1–12. [[CrossRef](#)]
37. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
38. Liu, M.; Sheng, L.; Yang, S.; Shao, J.; Hu, S.-M. Morphing and sampling network for dense point cloud completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
39. Wang, Y.; Tan, D.J.; Navab, N.; Tombari, F. Softpoolnet: Shape descriptor for point cloud completion and classification. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
41. Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R.R.; Hu, S.-M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [[CrossRef](#)]
42. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021.
43. Tatarchenko, M.; Richter, S.R.; Ranftl, R.; Li, Z.; Koltun, V.; Brox, T. What do single-view 3d reconstruction networks learn? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.

44. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. A papier-mâché approach to learning 3d surface generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
45. Zhang, J.; Chen, W.; Wang, Y.; Vasudevan, R.; Johnson-Roberson, M. Point set voting for partial point cloud analysis. *IEEE Robot. Autom. Lett.* **2021**, *6*, 596–603. [[CrossRef](#)]
46. Wen, X.; Li, T.; Han, Z.; Liu, Y.-S. Point cloud completion by skip-attention network with hierarchical folding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
47. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.