



Article

A Partial Point Cloud Completion Network Focusing on Detail Reconstruction

Ming Wei ^{1,2}, Jiaqi Sun ^{1,2}, Yaoyuan Zhang ^{1,2}, Ming Zhu ^{1,*}, Haitao Nie ¹, Huiying Liu ^{1,2} and Jiarong Wang ¹

¹ Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; weiming19@mails.ucas.ac.cn (M.W.); sunjiaqi19@mails.ucas.ac.cn (J.S.); zhangyaoyuan19@mails.ucas.ac.cn (Y.Z.); niehaitao@ciomp.ac.cn (H.N.); liuhuiying21@mails.ucas.ac.cn (H.L.); wangjiarong@cust.edu.cn (J.W.)

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

* Correspondence: zhuming@ciomp.ac.cn

Abstract: The point cloud is disordered and unstructured, and it is difficult to extract detailed features. The detailed part of the target shape is difficult to complete in the point cloud completion task. It proposes a point cloud completion network (BCA-Net) focusing on detail reconstruction, which can reduce noise and refine shapes. Specifically, it utilizes residual deformation architecture to avoid error points. The break and recombine refinement method is used to recover complete point cloud details. In addition, it proposes a bilateral confidence aggregation unit based on recurrent path aggregation to refine the coarse point cloud shape using multiple gating. Our experiments on the ShapeNet and Complete3D datasets demonstrate that our network performs better than other point cloud completion networks.

Keywords: point cloud completion; deep learning; point cloud processing; loop gating unit



Citation: Wei, M.; Sun, J.; Zhang, Y.; Zhu, M.; Nie, H.; Liu, H.; Wang, J. A Partial Point Cloud Completion Network Focusing on Detail Reconstruction. *Remote Sens.* **2023**, *15*, 5504. <https://doi.org/10.3390/rs15235504>

Academic Editor: Giuseppe Casula

Received: 25 September 2023

Revised: 15 November 2023

Accepted: 23 November 2023

Published: 25 November 2023



Copyright: © 2023 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

LiDAR acquires point clouds from the environment in three-dimensional vision. However, many uncertain environmental factors can affect the accuracy of the data. These include partial occlusion of targets and blurring of target motion. In addition, LiDAR systems often have low spatial resolution and may miss vital information about the environment [1–3]. As a result, raw point clouds collected by LiDAR can be sparse and incomplete, leading to significant differences from the actual geometry of objects and affecting the sensor system's perception of the environment [4,5]. To solve these problems, point cloud completion techniques can be utilized to reconstruct and restore the missing information in sparse and incomplete point clouds. The process involves using various algorithms to fill in gaps in the data and create a more realistic representation of the environment. By improving the accuracy of point cloud data, sensor systems can better perceive their surroundings and make more informed decisions [6]. Traditional point cloud completion methods mainly include the geometric symmetry method, surface reconstruction method, and template matching method. The geometric symmetry method is aimed at symmetrical objects, which can hardly be adapted to objects in nature. Surface reconstruction methods are also interpolation and fitting methods, which infer the location of missing points from existing points. However, missing locations are often uncertain and uneven. It can hardly fill in the shape accurately. And template matching method requires large databases, and the missing target must exist in the database. Not only the complexity is high, but also the universality is poor. The traditional method can hardly complete the unknown shape. Some partial shapes of the target are a lot of incomplete, which will bring great difficulties to the completion task. Therefore, the most popular point cloud completion method is to use the learning and optimization performance of deep learning networks to estimate the shape of the incomplete part [7].

Point cloud processing framework: In the completion task, the point cloud is disordered and unstructured, unlike a two-dimensional image. Normal convolution frameworks cannot be used in point cloud processing tasks. Therefore, different methods of framework have been proposed to solve the unstructured problem in the point cloud completion task. PCN [8] utilizes the multi-layer perceptron proposed by Point-Net [9] and directly maps part of the point cloud to the full shape through the encoding and decoding structure. It reduces intermediate steps and losses. It is also the most widely used point cloud processing framework. Since then, many scholars have proposed some other frameworks to process point clouds. GR-Net [10] maps the point cloud to the grid. They carry out three-dimensional convolution to the point cloud grid to retain the complete structure information. Similarly, the regularized point cloud is equivalent to being unitized and down-sampled and cannot retain the features of the detailed points. DGCNN [11] utilizes graph convolution on the restored topology to extract features from point clouds. However, its learning ability is limited, and local point clouds cannot be effectively restored. Top-Net [12] introduces a layered tree network that adapts to point cloud topology, focusing mainly on overall structure while neglecting details. ProxyFormer [13] designs a missing part sensitive transformer to transform the random normal distribution into reasonable position information. Predicted point agents are more sensitive to the characteristics and location of missing parts, and they are better suited for subsequent coarse-to-fine processing. USSPA [14] proposes an unsupervised symmetric shape-preserving automatic coding network. And, unlike previous methods of training each class separately, it can train multiple classes of data at once through classifier-guided discriminators. ACL-SPC [15] proposes a self-monitoring framework for point cloud completion. It takes a single partial input and attempts to output a complete point cloud using an adaptive closed-loop system that forces the same output for changes in the input. The challenge of recovering details is an important problem for almost all point cloud completion frameworks, as it determines the effectiveness of the completion process [16–18]. However, these frameworks need to transform or change the original information, and the comprehensive ability of the network is limited. Therefore, BCA-Net chooses the multi-layer perceptron framework to extract the point cloud features of the network, which has greater advantages in efficiency and cost. On this basis, it conducts special treatment for detailed reconstruction.

Residual structure: Residual structure can alleviate the problem of gradient disappearance caused by increasing network depth, which will contribute to the complete point cloud completion effect. Notable work, such as SA-Net [19], introduces a skip attention mechanism to generate complete point clouds with different resolutions by passing geometric information about local regions of incomplete point clouds. ASHF-Net [20] proposes a layered folding decoder with gated skip attention and multi-resolution completion targets to efficiently utilize partial input local structural details. NSFA [21] proposes two feature aggregation strategies to express and reconstruct coordinates from their combination. Therefore, BCA-Net proposes a residual deformation architecture to reduce the noise and focus on the feature learning process. Every path optimization may lead to a deviation in the learning direction. Therefore, it is necessary to maintain regularization. Features are constantly normalized in the convolution process to prevent perturbations caused by too many layers in the network. Many networks make use of it to achieve efficient feature extraction [22,23]. The residual deformation architecture can ensure the accuracy of information and prevent structural deviations during multi-scale changes [24,25].

Detailed reconstruction method: The single use of the point cloud processing framework is insufficient for point cloud completion tasks. It may implement a global shape estimate, but the details are ignored. Therefore, it is necessary to design detailed reconstruction modules to enhance the point cloud completion effect. PF-Net [26] proposes a multi-resolution encoder (MRE) to extract multi-layer features from partial point clouds and their low-resolution feature points. It enhances the ability of the network to extract semantic and geometric information. Folding-Net [27] proposes increasing the possibility of points by learning to fold two-dimensional grids into three-dimensional space. Based

on it, ECG [28] proposes an edges-aware feature extension module that up-samples or expands point features via graph convolution, which preserves local geometric details better than simply copying point features up-samples. Spare-Net [29] proposes channel-focused convolution of edge perception, which not only considers local information in K-neighbors but also makes wise use of the global context by aggregating global features and weighting each point's feature channel concerns accordingly. VRC-Net [30] adopts a two-subnetwork structure. One subnetwork uses a coarse complement point cloud generated by another subnetwork to enhance structural relationships through learning multi-scale local point features. It uses Point Self-focused kernel (PSA) and Point Selective Kernel Module (PSK) to further improve performance for detail recovery. Cascaded-PC [31] designs a lifting module to double the size of the points while refining the position of the points through feature shrinkage and expansion units. LAKe-Net [32] designs a refinement subnet to feed the multi-scale surface skeleton to each recursive skeleton auxiliary refinement module during the complement process. Continuous optimization of the point route is effective in restoring shape details. PMP-Net [33] proposed an RPA module based on the original GRU [34,35] to memorize and aggregate the route sequence of points and obtain the exact position by continuously refining the moving path. Path aggregation is an efficient way to optimize point cloud details. However, the independent control of the reset gate in the polymerization unit will affect the polymerization effect. The aggregation unit can be optimized, and the effect needs to be enhanced.

Therefore, to further improve the point cloud completion performance in detail, it proposes a bidirectional confidence aggregation unit in the network. By adding a confidence gate to the update gate and the reset gate, BCA-Net can continuously standardize the reliability of points. It can strengthen the context information and guide the point cloud details of network completion. Because error points and noise points are easily occur in the entire process of point cloud compensation, not all point reliability is consistent. It considers the confidence levels of updates and resets when moving path predictions, rather than being confident that all points are correct. The confidence gate allows for more accurate and reliable point route refinement, resulting in better point cloud detail recovery.

In addition, it proposes a break and recombine refinement module to further enhance the point cloud detail reconstruction in the network. It is designed to assist in the recovery of point cloud details by fusing features at a deep level. Unlike traditional feature fusion methods that often require additional processing and optimization, the break and recombine refinement module allows for internal information processing by converting points from the optimization part into a new dimension. In image processing, a common practice is to convert the time domain image to the frequency domain for processing. It deletes or adds information in the frequency domain and then converts it back to the time domain. It enables internal information processing. From this point of view, it breaks and reorganizes the new points and completion points in the optimization part, and it combines one-dimensional and two-dimensional convolution. Fusion takes place at a deep level. Points are fused in a new dimension. In addition, the attention mechanism is used to weight the processing, making more important parts highlighted, and multiple branches are added to enrich the fusion level [36,37]. The structure is inspired by CC-Net [38]. In BCA-Net, it built the breaking recombination module to enhance the fusion effect between points. It makes the point cloud details more precise.

The main contributions of the paper are summarized as follows:

1. It proposes a residual deformation architecture to regulate the learning direction of the network and reduce noise, ensuring the accuracy of the structure.
2. It designs the break and recombines refinement for high dimensional internal processing, achieving deep fusion and optimization, and recovering point cloud details.
3. It designs a bidirectional confidence aggregation unit to guide the recovery of point details by considering the confidence levels of updates and resets during moving path predictions.

4. The experiments demonstrate that our network has an enhanced effect on details and suppressed noise, achieving effective end-to-end point cloud completion.

2. Methods

Our network is mainly divided into two phases: residual deformation architecture and break and recombine refinement. Figure 1 shows the process of the input point cloud from coarser to finer. In phase (a), the input point cloud is extracted and down-sampled by the set abstraction module. The residual structure is adopted in the feature propagation process, which retains the original characteristics of the network and prevents gradient explosion. It will be introduced in Section 2.1. In addition, it designs a bilateral confidence aggregation unit to enhance local details and obtain coarse point clouds, which will be introduced in Section 2.2. In phase (b), it designs the break and recombine module for further refinement to obtain the final complete point cloud, which will be introduced in Section 2.3.

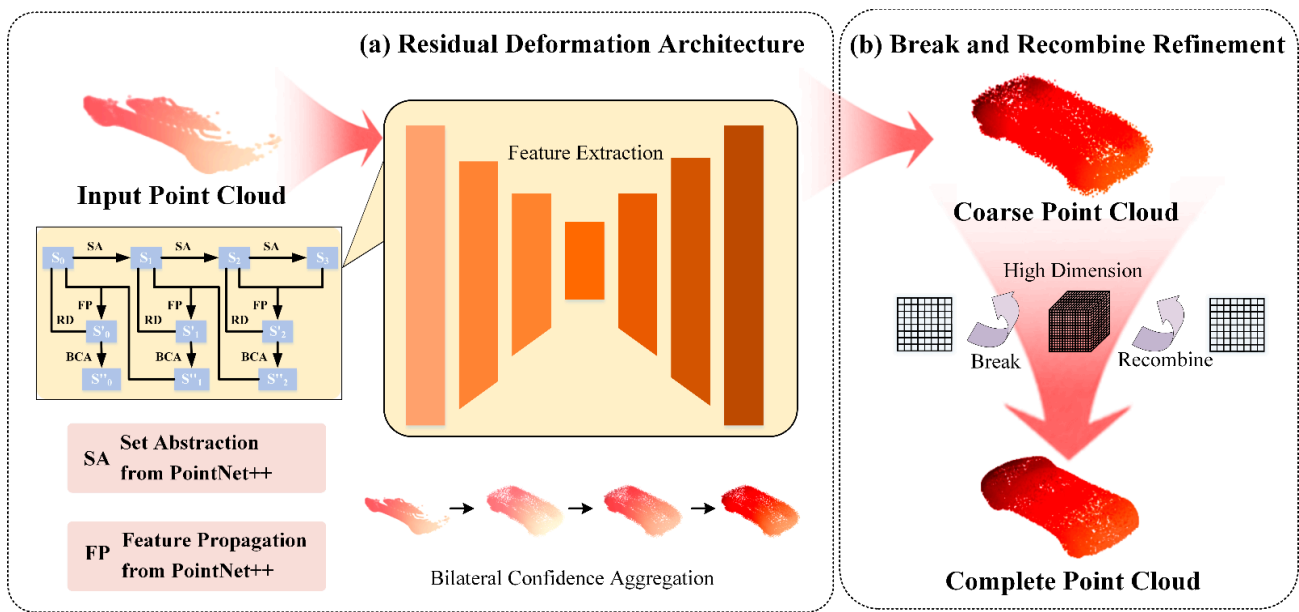


Figure 1. Overall Architecture. The network consists of (a) (left dashed box) and (b) (right dashed box). The detailed structure of the residual deformation architecture is shown in the box in the middle of the left. The trailing arrow indicates the direction of the network. S represents the scale change process in the process of feature extraction and propagation. RD is residual deformation architecture. SA is set abstraction and FP is feature propagation. BCA is the bilateral confidence aggregation unit.

2.1. Residual Deformation Architecture

As shown in the lower-left corner of Figure 1a, it designs a residual deformation architecture to extract the features of the fragmentary point cloud. It connects the point cloud of the previous stage through the RD layer. The residual structure can regulate the direction of the network and reduce the noise. If the scale of the feature is $n \in 0, \dots, N$, the output result can be expressed as

$$S'_n = RD(S_n, FP(S_n, S''_{n+1})) \quad (1)$$

$$S''_n = BCA(S'_n) \quad (2)$$

The set abstraction (SA) layer in the network abstracts a set of points into a set of fewer points and constructs a local region, using the centroid neighborhood grouping. The local region is then encoded into an eigenvector, which helps compensate for the loss of precision caused by maximum pooling. The feature propagation (FP) layer utilizes an inverse distance weighted average based on K-nearest neighbor to interpolate features

point by point. It connects features to the original point set through the unit point network. The diagram of the *SA* module and *FP* module is shown in Figure 2, which are all from PointNet++ [39]. The output result of the structure is a coarse point cloud S_0'' .

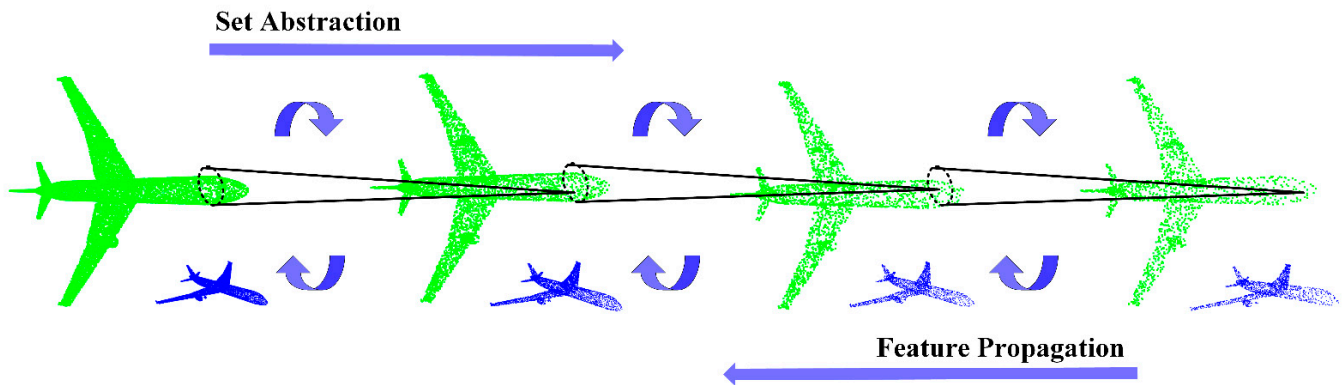


Figure 2. The construction of set abstraction and feature propagation modules. Aircraft point clouds are constantly grouped and sampled from left to right. And they are constantly propagated and interpolated from right to left. It shows the top and side views.

The residual deformation is used in the feature extraction step. Its purpose is to avoid deviating from the right direction due to the deepening of the number of network layers while extracting point cloud features. In the whole feature extraction process, the point cloud first passes through the set abstraction (*SA*) layer to reduce the resolution, and then through the feature propagation (*SA*) layer to improve the resolution. In the process of improving the resolution, the fusion of the original point cloud features with the same resolution is required. Our residual deformation further actively guides the direction of the convolution to optimize the features after fusion. It can reduce the loss caused by resolution changes.

2.2. Bilateral Confidence Aggregation Unit

As the network gradually replenishes the missing point cloud, the number of points will multiply. The accuracy of the added points is ignored. Once the optimization direction is wrong, noise points and discrete points will accumulate. To solve the problem, PMP-Net proposes a recurrent path aggregation (*RPA*) module. Further, the paper proposes the *BCA* module to further determine the reliability of the reset gate by adding a new gate control. The optimization for point paths changes the optimization idea and gradually optimizes the position of the point through a point path search. The network can plan the learning path for noise and error points. The point path search method can plan the learning path of noise points and error points. It utilizes a coarse-to-fine route for iterative learning. It is based on *RGU*, which is a kind of recurrent neural network.

Recurrent neural network (*RNN*) has excellent performance in processing sequence data. Its core concept is the cell state and gate structure, which can combine the front and back inputs across time. Therefore, it can be used as an iterative unit of the point path. The cell state is equivalent to a path that can transmit relevant information, and the feature vector can be passed on as a memory. Gate structure determines whether information needs to be remembered or forgotten. Figure 3 shows a comparison of our bilateral confidence aggregation unit with other similar modules. Each iteration structure retains both the output and the hidden state as the path optimization direction of the current learning. And it is passed to the next iteration process. gated recurrent unit (*GRU*) is a type of *RNN* and is widely used because they are easy to train and have high performance. As shown in Figure 3a, it has two gates: the reset gate and the update gate. It utilizes the sigmoid function as the gating signal. The information control signal is between 0 and 1, with values closer to 0 indicating that the information should be discarded and values closer to 1 indicating that the information should be retained.

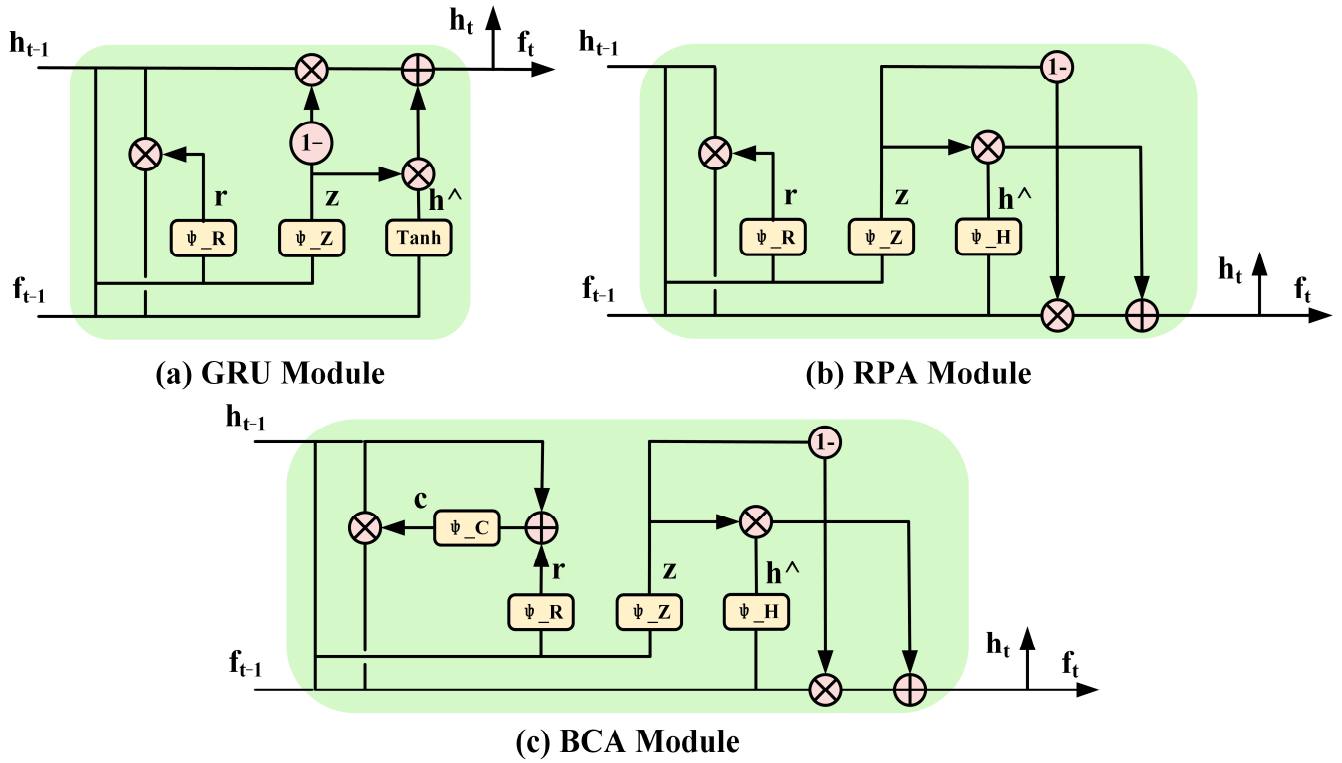


Figure 3. The construction of the bilateral confidence aggregation unit compared to other loop gating modules. (a) is the gated recurrent unit (GRU) module, (b) is the recurrent path aggregation (RPA) module, and (c) is our bilateral confidence aggregation (BCA) module.

The reset gate selectively combines input information with memory information to determine the degree of retention of information from the previous moment. The reset gate output at time t can be expressed as

$$r_t = \sigma_{sig}(W_r \cdot [h_{t-1}, f_{t-1}]) \quad (3)$$

where σ_{sig} is the sigmoid function. W_r is the weight matrix of the reset gate. h_{t-1} is the hidden state at time $t - 1$. f_{t-1} is the input at time $t - 1$.

The update gate can determine which information should be discarded or retained and select the updated information to be output. The update gate output z_t at time t can be expressed as

$$z_t = \sigma_{sig}(W_z \cdot [h_{t-1}, f_{t-1}]) \quad (4)$$

where W_z is the weight matrix of the update gate.

The candidate's hidden state \hat{h} can be expressed as

$$\hat{h}_t = \sigma_{tanh}(W_h \cdot [r_t \cdot h_{t-1}, f_{t-1}]) \quad (5)$$

where W_h is the weight matrix of the hidden state. σ_{tanh} is the tanh function.

PMP-Net introduced the gated recurrent unit (GRU) module into the point cloud completion network as a path search optimization unit. They design a recurrent path aggregation (RPA) module which is shown in Figure 3b. PMP-Net proves that the module has a better effect than the original GRU module through experiments. However, the sequence processing needs high semantic requirements, so the need for hidden states is higher. The main body of point path search is the original input path, and the hidden state is used to assist path optimization. In addition, the gradient of the sigmoid function and tanh function is close to 0 near the extreme value, and the optimization algorithm updates the network slowly. As a result, the Relu functions are better suited for deep networks.

Like GRU modules, RPA modules include a reset gate and an update gate. The difference is that the candidate's hidden state \hat{h} is expressed as

$$\hat{h}_t = \sigma_{relu}(W_h \cdot [r_t \cdot h_{t-1}, f_{t-1}]) \quad (6)$$

where σ_{tanh} is the Relu function.

The output result of the time f_t can be expressed as

$$f_t = (1 - z_t) \cdot [r_t \cdot h_{t-1}, f_{t-1}] + z_t \cdot \hat{h}_t \quad (7)$$

It designs the bilateral confidence aggregation (BCA) unit to further improve the gating performance. As shown in Figure 3c, it designs a new confidence gate after the reset gate to control the confidence of the network to the previous hidden state, rather than directly introducing it. The confidence gate quantifies the uncertainty of the hidden state and measures the confidence of path movement. It is very effective in restoring shape details. Unit mainly consists of two branches, the hidden state of the previous moment is the top branch, and the input of the previous moment is the bottom branch. In the RPA structure, the control for both the reset and update doors comes from the direct fusion of the top and bottom branches, ultimately controlling the aggregation of the bottom branches. In this process, the effect of the top branch is not obvious, and the hidden state is directly exploited without processing. In the BCA module, the hidden state is first further aggregated and restricted, the top branch is effectively used. The module realizes the bidirectional aggregation of the top and bottom branches.

The confidence gate output c_t at time t can be expressed as

$$c_t = \sigma_{sig}(W_c \cdot [h_{t-1}, r_t]) \quad (8)$$

The candidate's hidden state \hat{h} is expressed as

$$\hat{h}_t = \sigma_{relu}(W_h \cdot [c_t \cdot h_{t-1}, f_{t-1}]) \quad (9)$$

The output result f_t can be expressed as

$$f_t = (1 - z_t) \cdot [c_t \cdot h_{t-1}, f_{t-1}] + z_t \cdot \hat{h}_t \quad (10)$$

2.3. Break and Recombine Refinement

After several iterations, the raw point cloud is completed into the coarse point cloud. As shown in Figure 4, it designs the break and recombine refinement module to further add detail points and optimizations to the point cloud. Breaking means increasing dimension, and recombine means decreasing dimension. Our fusion optimization takes place in high-dimensional space. To better show the structure, it utilizes the dimensions that represent the features. It obtains the coarse point cloud $C \in \Psi^{B,C,N_1}$ with N_1 points through the previous step and sets the point $A \in \Psi^{B,C,N_2}$ with N_2 points to be fused. It utilizes the method of initial concatenation and then fusion. The concatenation result of the two sets of data C_{ori} can be expressed as

$$C_{ori} = cat(C, A) \quad (11)$$

Next, the splicing results are broken down into smaller parts. During the breaking process, dimensions are increased, and internal information is revealed. It allows for a more detailed analysis of the point cloud structure. The goal of the step is to reveal the hidden relationships and patterns within the point cloud, which can then be used for further processing and analysis. The high-dimensional fusion point cloud P can be represented as

$$P \in \Psi^{B,C,P_i,P_j} = AD(C_{ori} \in \Psi^{B,C,N_1+N_2}, \dim) \quad (12)$$

where B is the batch size, C is the number of channels, and \dim is the broken dimension. P_i and P_j are the result of the point cloud dimension expansion. AD is the ascending dimension.

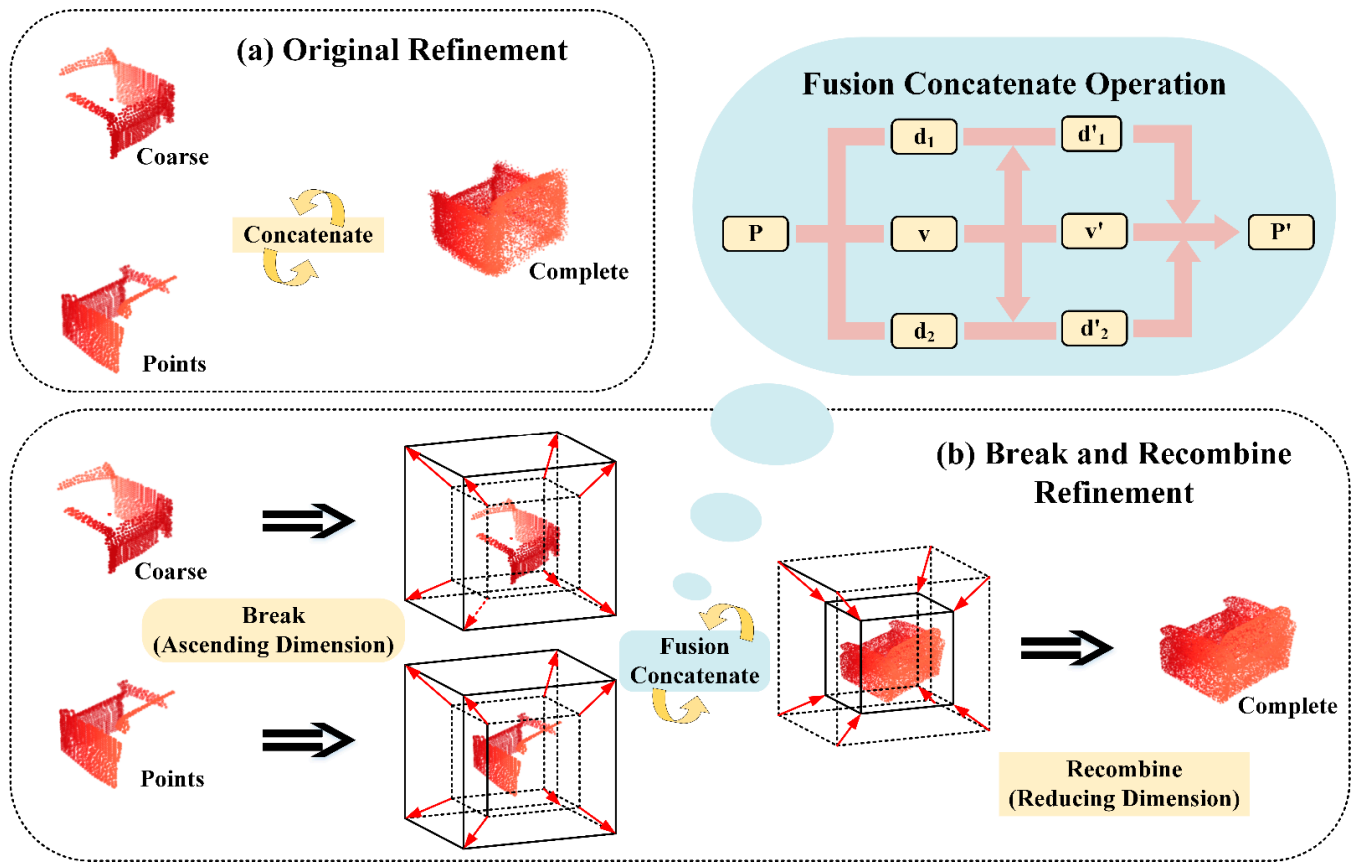


Figure 4. The construction of break and recombine refinement: (a) is the original optimization structure; (b) is our break and recombine refinement structure.

It optimizes the result of raising dimensions in higher dimensions. To better integrate internal information, it sets up a structure of one main road and two auxiliary branches. The three routes can be expressed as $v \in \Psi^{B,C/2,P_i,P_j}$, $d_1 \in \Psi^{B,C/2,P_i,P_j}$, and $d_2 \in \Psi^{B,C/2,P_i,P_j}$.

It constructs a cross-multiplicative structure to propagate the features of the main road to the branch road, which can be obtained as

$$\begin{cases} v_1 \in \Psi^{BP_j,C/2,P_i} = view(v \in \Psi^{B,C/2,P_i,P_j}) \\ v_2 \in \Psi^{BP_i,C/2,P_j} = view(v \in \Psi^{B,C/2,P_i,P_j}) \\ \hat{d}_2 \in \Psi^{BP_i,P_j,P_j} = view(d_2 \in \Psi^{B,C/2,P_i,P_j}) \\ \hat{d}_1 \in \Psi^{BP_j,P_i,P_i} = view(d_1 \in \Psi^{B,C/2,P_i,P_j}) \end{cases} \quad (13)$$

where *view* is a dimensional reconstruction operation.

The initial result of the branch d'_1 and d'_2 can be expressed as

$$d'_1 \in \Psi^{B,C/2,P_i,P_j} = bmm(v_1, view(v_1, \hat{d}_1)) \quad (14)$$

$$d'_2 \in \Psi^{B,C/2,P_i,P_j} = bmm(v_2, view(v_2, \hat{d}_2)) \quad (15)$$

where *bmm* is matrix multiplication.

Finally, it carries out the fusion of each branch and gets the feature confidence *f* that can be expressed as

$$f = \sigma_{sig}(W(cat(v, d'_1, d'_2))) \quad (16)$$

where W is the weight matrix. Feature confidence is used to focus on the more important part of the feature to be fused. The fusion result P' can be expressed as

$$P' = f \times P + P \quad (17)$$

2.4. Loss Function

It utilizes Chamfer Distance and Earth Mover's Distance as loss functions to train the network. Chamfer Distance measures the average distance between each point in a point cloud and the nearest point in another point cloud. It is the most extensive way to judge the completion effect. For point clouds S_1 and S_2 , the CD is defined as

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\| + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\| \quad (18)$$

EMD is a definition of distance measurement that can be used to measure the distance between two distributions, which can be expressed as

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \|x - \phi(x)\| \quad (19)$$

where ϕ is considered as a bijection that minimizes the average distance between corresponding points in S_1 and S_2 .

The total loss function is defined as

$$L = d_{CD} + d_{EMD} \quad (20)$$

3. Results

3.1. Setup

To verify the effectiveness of the network, it conducts comparative experiments on the network on the ShapeNet dataset and the Complete3D dataset, which are described in Section 3.2. Ablation experiments are performed on the three proposed modules on the ShapeNet dataset, which are described in Section 3.2.

ShapeNet: The ShapeNet dataset is from PCN in our work. The original dataset consisted of 30,974 3D models. Using a uniform sampling method, they back-projected 16,384 sample points of the target into 3D to generate partial point clouds. Some of the point clouds have different points in some shapes. Each complete shape contains 16,384 points. The dataset includes eight categories such as aircraft, watercraft, and so on. It follows the same dataset segmentation method as the PCN for fair comparison.

Complete3D: The complete3D dataset is a large-scale 3D object dataset derived from the ShapeNet dataset, created by Top-Net. The number of points in both partial and full point clouds is 2048 points. Each complete model corresponds to only one complete point cloud. It follows the same dataset segmentation to make fair comparisons.

In addition, our network is designed through the PyTorch framework. An NVIDIA GTX 2080 TI GPU is used in the experiment. The batch size is set to 24. To make a fair comparison with other networks, it only focuses on the innovation of network structure, but does not improve the training method. The optimizer is Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ during training, which is efficient and stable. The initial learning rate is set to 10^{-3} , and the epoch index decays by 0.5 every 20 epochs.

3.2. Results of Comparative Experiments

3.2.1. ShapeNet

It first experiments on the ShapeNet dataset, which is widely used. It can compare it more clearly with other network results. The effect of our network compared with other networks on the ShapeNet dataset is shown in Table 1. To make a clear comparison, the

common L2 chamfer distance used by other networks was chosen as the evaluation method. It is the CD in Equation (18) averaged by the point number.

Table 1. Point cloud completion on ShapeNet dataset in terms of per-point L1 Chamfer Distance $\times 10^3$. The bold numbers are the best.

Model	Folding-Net	Top-Net	PCN	GR-Net	PMP-Net	BCA-Net
Airplane	9.49	7.61	5.50	6.45	5.50	5.19
Cabinet	15.80	13.31	22.70	10.37	11.10	10.79
Car	12.61	10.90	1063	9.45	9.62	9.51
Chair	15.55	13.82	8.70	9.41	9.47	9.13
Lamp	16.41	14.44	11.00	7.96	6.89	6.62
Sofa	15.97	14.78	11.34	10.51	10.74	10.95
Table	13.65	11.22	11.68	8.44	8.77	8.03
Watercraft	14.99	11.12	8.59	8.04	7.19	7.26
Overall	14.31	12.15	9.64	8.83	8.66	8.43
Time	-	-	-	0.020	0.016	0.018

In five of the eight categories, our results are better. PMP-Net utilizes the same framework as our network, and our results are better than it. GR-Net is based on a 3D convolution framework that consumes more time and space costs. Our framework is faster than 3D convolutional frameworks. In addition, the 3D convolution framework uses the grid as the intermediate quantity and voxelizes the point cloud, which does not lose the global structural features. Therefore, other frames may have an advantage when it comes to completing the overall shape. As shown in the table, BCA-Net has the best overall results. However, the results are slightly lower than GR-Net in the cabinet, car, and sofa categories. It is because the objects of these three categories are holistic. The target points are relatively concentrated and the requirements for details are lower. BCA-Net is more focused on restoring details and has more advantages in practical applications. The overall shape has been preliminarily estimated, and the details are more important in some operations that require three-dimensional information.

As shown in Figure 5, the first row is the aircraft category, where the wing shape is better recovered from a complete absence. The second line is the lamp category, where the bracket section is less noisy and closer to ground reality. The third row is for the car category, where the outline of the wheels and partial details of the windshield are more prominent, and the last line is the table category; our network has a much more nuanced grasp of the breadth of detail. The performance of BCA-Net is excellent, especially when it comes to detail reconstruction.

3.2.2. Complete3D

Since the ground truth of the Complete3D dataset has a smaller number of points, it conducts another experiment on the Complete3D dataset to further verify the effect of the network. To make a clear comparison, the common L2 Chamfer Distance used by other networks was chosen as the evaluation method. It is the CD replacing the L1-norm in Equation (18) with the L2-norm. For point clouds S_1 and S_2 , the CD is defined as

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \quad (21)$$

As shown in Table 2, Unlike the results in the previous section, BCA-Net is better than GR-Net in all categories when targeting many missing points. It proves that frameworks with intermediate-form transformations have greater losses in this case. The overall shape of the original information is very important to them. When it is missing, it is less effective. In addition, BCA-Net is unfortunately not optimal in the Cabinet, Car, and Table categories. It is more suitable for detail reconstruction when the number of missing points is small. However, our network was able to achieve the best results overall in the comparison exper-

iment on Complete3D, reaching the optimal value in most categories. It demonstrates the effectiveness of our method and its ability to handle large-scale datasets like Complete3D.

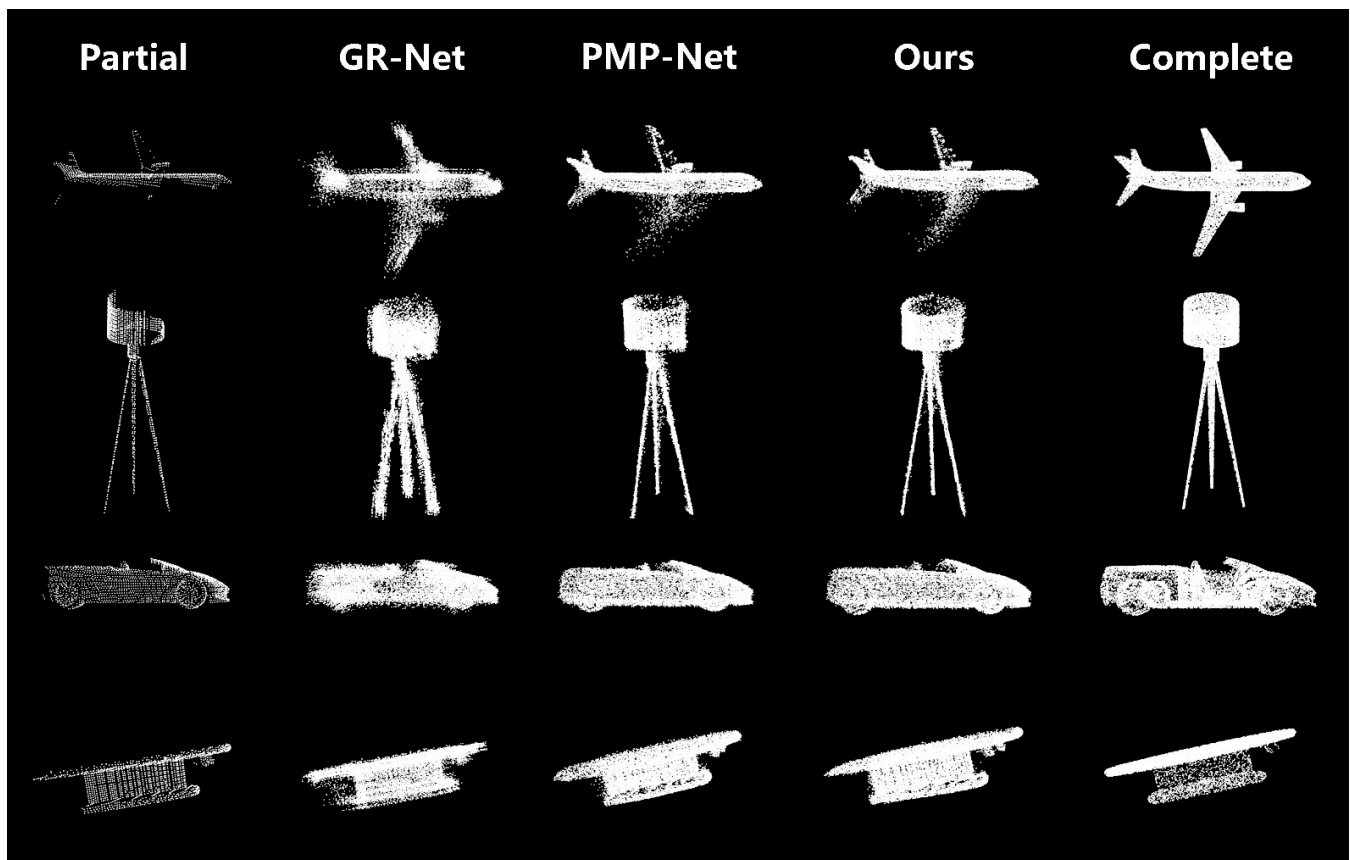


Figure 5. The results of visual on ShapeNet dataset.

Table 2. Point cloud completion on Completion3D dataset in terms of per-point L2 Chamfer distance $\times 10^4$. The bold numbers are the best.

Model	Folding-Net	PCN	Top-Net	SA-Net	GR-Net	CRN	PMP-Net	BCA-Net
Airplane	1.28	0.98	0.73	0.53	0.61	0.40	0.39	0.26
Cabinet	2.34	2.27	1.88	1.45	1.69	1.32	1.47	1.35
Car	1.49	1.24	1.29	0.78	0.83	0.83	0.86	0.71
Chair	2.57	2.51	1.98	1.37	1.22	1.06	1.02	1.20
Lamp	2.18	2.27	1.46	1.35	1.02	1.00	0.93	0.87
Sofa	2.13	2.03	1.63	1.42	1.49	1.29	1.24	0.99
Table	2.07	2.03	1.49	1.18	1.01	0.92	0.85	1.35
Watercraft	1.15	1.17	0.88	0.88	0.87	0.58	0.58	0.49
Overall	1.91	1.82	1.45	1.12	1.06	0.92	0.92	0.90

3.3. Results of Ablation Experiments

It proposes three innovative modules in the paper. To prove the role of the modules, it conducts ablation experiments on the ShapeNet dataset.

3.3.1. Residual Deformation Architecture

Residual deformation architecture is proposed in the feature extraction step of the network. The multi-resolution structure is conducive to collecting information at various scales. Table 3 shows the effect of adding residual deformation architectures to our network. It conducts four sets of comparison experiments, without adding residual deformation structure, adding one layer, adding two layers, and adding three layers. “None” represents

the result without adding any structure. “+Res*1” represents the result of adding residual deformation architecture of one layer. “+Res*2” represents the result of adding residual deformation architecture of two layers. “+Res*3” represents the result of adding residual deformation architecture of three layers. It uses the Chamfer Distance (CD) to verify the effect of this structure. The relationship between the number of layers and the effect of residual deformation can be measured from the experimental results. The network effect gradually improves with the addition of residual deformed architectures. However, the time increased by less than 0.1ms with each additional layer. It demonstrates the simplicity and efficiency of our structure. In addition, as shown in Figure 6, the Residual Deformation Architecture designed by us pays more attention to the original information, effectively suppressing stray points and noise points.

Table 3. Ablation experiment of residual deformation architecture in terms of per-point L1 Chamfer Distance $\times 10^3$. The bold numbers are the best.

Model	Overall CD	Time (ms)
+None	8.64	1.6
+Res*1	8.62	1.6
+Res*2	8.61	1.7
+Res*3	8.60	1.7

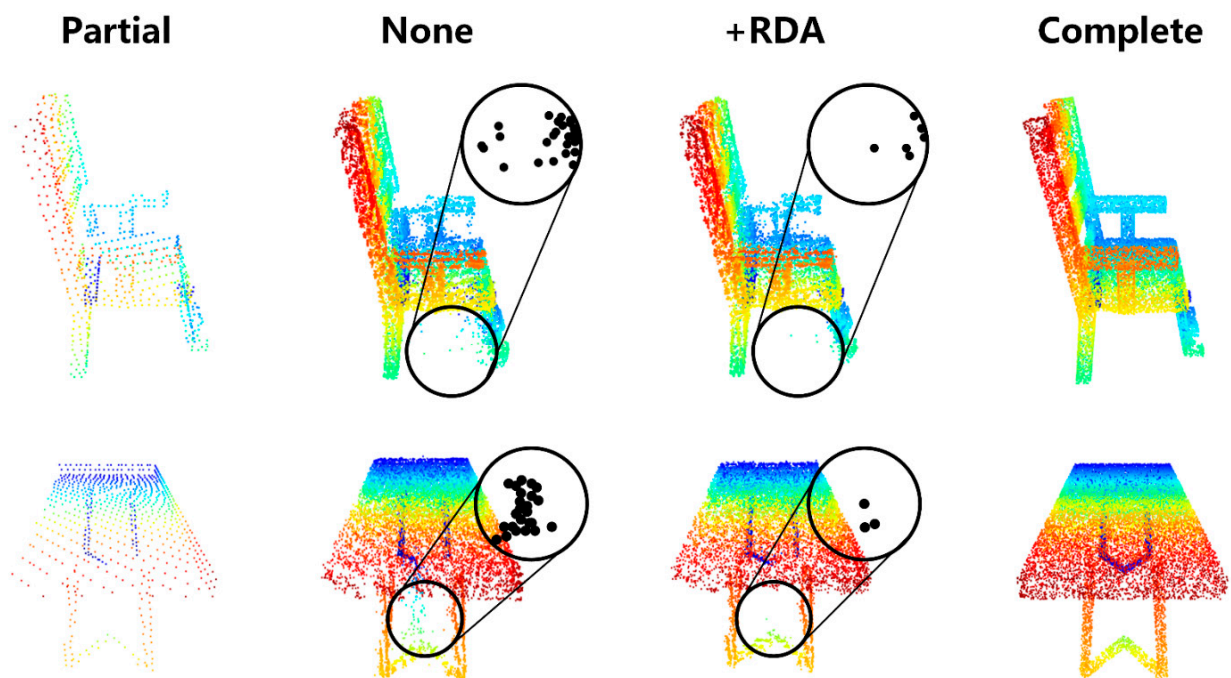


Figure 6. Visualization effect of residual deformation architecture. details in the smaller circle are highlighted in the larger circle. Noise points are magnified in large circles.

3.3.2. Bilateral Confidence Aggregation Unit

Bilateral Confidence Aggregation Unit increases the confidence gate which can be well restored to a more realistic detailed shape. As shown in the dashed box on the left of Figure 7, after adding the module, the lamp head shape was better reconstructed and had clearer boundaries. The actual shape of the bottom of the lamp in the dotted frame on the right is a vertical column. The network estimated the wrong shape before adding the module. As a rule of thumb, since the dataset consists of eight different types, the network recognizes the lamp type as something else, such as the chair. On the contrary, after the addition of modules, the network was more resilient to details, and the network successfully recognized categories and reconstructed details. In addition, the partial point

cloud of the left leg part is missing in the lower dotted line frame. After adding the module, its shape details are effectively completed, which is closer to the real shape.

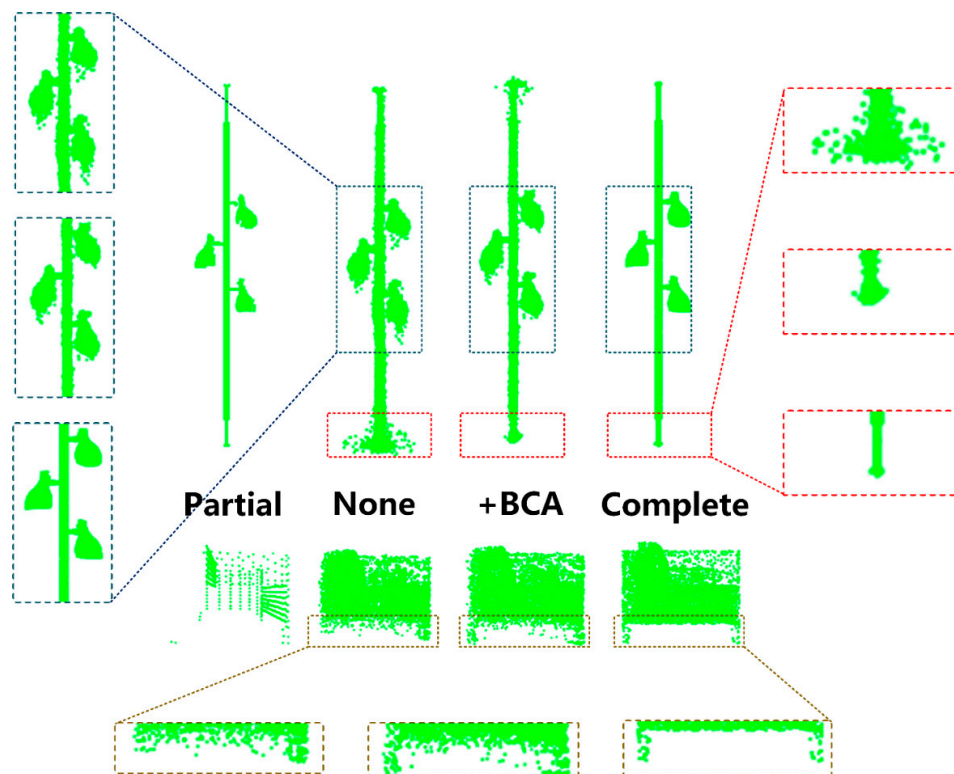


Figure 7. Visualization effect of bilateral confidence aggregation unit. The top is the lamp, and the bottom is the side of the sofa. The details are highlighted in a rectangular dotted box.

Table 4 shows the ablation experiments of the BCA unit and the comparison of the effect of adding confidence gates in different positions. “Ori” represents the result without adding any module. “+GRU” represents the result of adding the GRU module. “+RPA” represents the result of adding the RPA module. “+BCA” represents the result of adding a confidence gate between the input hidden state and the reset gate based on the RPA module, which is the BCA module. As shown in the results of the experiment in Table 4, adding a path aggregation module has a beneficial effect. When it adds a confidence gate between the input hidden state and the reset gate, the confidence gate constrains the relationship between the hidden state and the reset. Point cloud details are better recovered.

Table 4. Ablation experiment of bilateral confidence aggregation in terms of per-point L1 Chamfer Distance $\times 10^3$. The bold numbers are the best.

Model	Air-Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Water-Craft	Overall
Ori	6.05	11.13	9.62	9.53	7.50	10.82	8.36	7.42	8.72
+GRU	5.37	11.00	9.71	9.48	6.82	11.16	8.25	7.39	8.65
+RPA	5.26	11.04	9.54	9.37	6.85	11.21	8.20	7.33	8.60
+BCA	5.34	10.81	9.57	9.34	6.84	11.14	8.22	7.41	8.58

3.3.3. Break and Recombine Refinement

Break and recombine refinement can enhance local effects through high dimensional fusion. Because our refinement module focuses more on detailed shapes, it uses FS to evaluate it. F-Score (FS) can show the ability to recover the detailed shape of the point cloud. It is defined as

$$FS = \frac{2P(d)R(d)}{P(d) + R(d)} \tag{22}$$

where P is precision. It calculated the percentage of corresponding points in the reconstructed point clouds within a certain threshold distance from the ground truth value. R is the recall rate. It calculated the percentage of the ground-truth value corresponding to the correct point within a certain threshold with the reconstructed point cloud distance.

As shown in Table 5, the break and recombine refinement module has an excellent effect on our network. The threshold of FS is set to 0.01. “None” represents the result without adding any module. “+BR” represents the result of adding the break and recombine refinement module. In all categories, the target point cloud is closer to ground truth. The CD decreased by 0.15 and the F-score increased by 0.01. Therefore, the break and recombine refinement module is effective in improving accuracy.

Table 5. Ablation experiment of break and recombine refinement in terms of per-point L1 Chamfer distance $\times 10^3$ (The lower is the best) and F-Score (The higher is the best). the bold numbers are the best.

	Model	Air-Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Water-Craft	Overall
CD	None	5.34	10.81	9.57	9.34	6.84	11.14	8.22	7.41	8.58
	+BR	5.19	10.79	9.51	9.13	6.62	10.95	8.03	7.26	8.43
FS	None	0.820	0.534	0.540	0.622	0.766	0.508	0.688	0.719	0.650
	+BR	0.836	0.543	0.552	0.634	0.773	0.513	0.700	0.730	0.660

The visualization is shown in Figure 8. In the aircraft category, the top half of the wing is almost completely missing, which means the network needs to reconstruct the shape of the wing through optimization. As shown in the detailed enlarged image on the left, points estimated from the initial network are scattered. However, the points are grouped to form an exact shape after adding the module. The shape of the left wing of the aircraft is closer to reality.

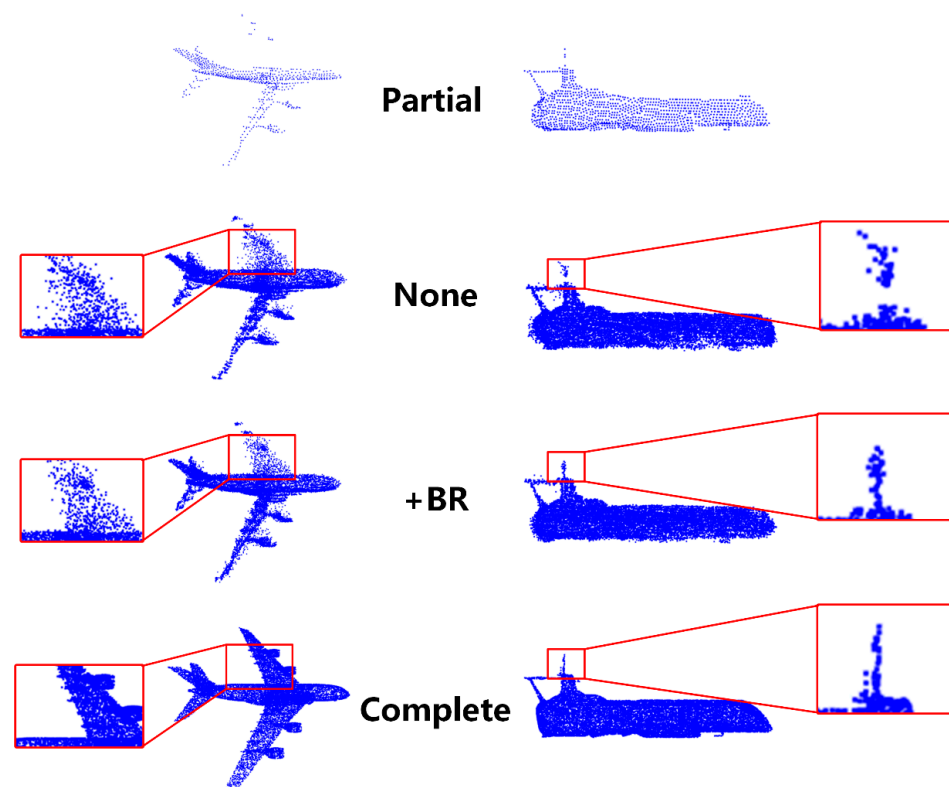


Figure 8. Visualization effect of break and recombine refinement module. The visualization and enlarged details of the aircraft are on the left. The visualization and enlarged details of the watercraft are on the right.

In addition, the top left corner shape is incomplete in the watercraft category. The detail area and the ship are connected. The completion point cloud is still disconnected from the hull before adding the module. After adding the module, the gap in the middle is filled and the details are more precise.

Table 6 shows the time consumed by the various modules. “None” represents the result without adding any module. “+RDA” represents the result of adding the residual deformation architecture. “+BCA” represents the result of adding the bilateral confidence aggregation module. “+BR” represents the result of adding the break and recombine refinement module. “ Δt ” represents the time increment. Our modules are simple and efficient. The modules proposed in the paper take about 1 ms. It achieves detailed recovery of the fragmentary point cloud with a mere 0.018 s running time.

Table 6. Comparison experiment of running time of each model.

Model	None	+ RDA	+BCA	+BR
Δt (s)	-	0.001	0.001	<0.001
Time (s)	0.016	0.017	0.018	0.018

4. Conclusions

This paper designs an end-to-end point cloud completion network (BCA-Net) to complete the shape of the partial point cloud. It can be used to improve the effectiveness of three-dimensional information in areas such as autonomous driving and smart industry. In addition, compared to other networks, our network can recover point cloud details better, which is vital in real complex applications. In this network, to solve the problem of noise points as the network deepens, it adopts the residual deformation architecture. It helps to avoid adding stray points and maintain the integrity of the point cloud. To further refine the recovered point cloud, it incorporates a break and recombine refinement module to the fusion structure. In addition, it constructs a bidirectional confidence aggregation unit to achieve progressive shape recovery. Our network has demonstrated exceptional performance in terms of accuracy and speed.

In addition, there are some disadvantages to the research. The network pays more attention to detail. Therefore, it is not perfect for targets with less detail and denser points. The target objects in the current datasets also have limitations in terms of variety and quantity. The universality of the method needs to be further improved. In the future, it will conduct research around more diverse and challenging targets.

Author Contributions: M.W. and Y.Z. contributed to the theory research and the conception of the experiments; M.W., J.S. and H.N. performed the experiments and analyzed the results; M.W. and H.L. wrote the paper and created the diagrams; M.Z. and J.W. contributed to scientific advising and proofreading. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Department of Jilin Province, China under grant number 20210201137GX.

Data Availability Statement: Data are contained within the article.

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

References

1. Xia, Y.; Xia, Y.; Li, W.; Song, R.; Cao, K.; Stilla, U. ASFM-Net: Asymmetrical Siamese Feature Matching Network for Point Completion. *arXiv* **2021**, arXiv:2104.09587v2.
2. Cai, Y.; Lin, K.Y.; Zhang, C.; Wang, Q.; Wang, X.; Li, H. Learning a Structured Latent Space for Unsupervised Point Cloud Completion. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5533–5543.
3. Huang, H.; Chen, H.; Li, J. Deep Neural Network for 3D Point Cloud Completion with Multistage Loss Function. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 4604–4609.
4. Manuele, S.; Gianpaolo, P.; Francesco, B.; Tamy, B.; Paolo, C. High Dynamic Range Point Clouds for Real-Time Relighting. *Comput. Graph. Forum* **2019**, *38*, 513–525.

5. Gurumurthy, S.; Agrawal, S. High Fidelity Semantic Shape Completion for Point Clouds Using Latent Optimization. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1099–1108.
6. Boulch, A.; Marlet, R. POCO: Point Convolution for Surface Reconstruction. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 6292–6304.
7. Chen, Z.; Long, F.; Qiu, Z.; Yao, T.; Zhou, W.; Luo, J.; Mei, T. AnchorFormer: Point Cloud Completion from Discriminative Nodes. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 13581–13590.
8. 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; pp. 728–737.
9. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.
10. Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Sun, W. GRNet: Gridding Residual Network for Dense Point Cloud Completion. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
11. 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*, 146. [[CrossRef](#)]
12. Tchapmi, L.P.; Kosaraju, V.; Rezatofghi, H.; Reid, I.; Savarese, S. TopNet: Structural Point Cloud Decoder. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 383–392.
13. Li, S.; Gao, P.; Tan, X.; Wei, M. ProxyFormer: Proxy Alignment Assisted Point Cloud Completion with Missing Part Sensitive Transformer. *arXiv* **2023**, arXiv:2302.14435.
14. Ma, C.; Chen, Y.; Guo, P.; Guo, J.; Wang, C.; Guo, Y. Symmetric Shape-Preserving Autoencoder for Unsupervised Real Scene Point Cloud Completion. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 13560–13569.
15. Hong, S.; Yavartanoo, M.; Neshatavar, R.; Lee, K.M. ACL-SPC: Adaptive Closed-Loop System for Self-Supervised Point Cloud Completion. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 9435–9444.
16. Zhang, J.; Zhang, H.; Vasudevan, R.; Johnson-Roberson, M. HyperPC: Hyperspherical Embedding for Point Cloud Completion. *arXiv* **2023**, arXiv:2307.05634.
17. Tan, H. Visualizing Global Explanations of Point Cloud DNNs. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 4730–4739.
18. Li, S.; Ye, Y.; Liu, J.; Guo, L. VPRNet: Virtual Points Registration Network for Partial-to-Partial Point Cloud Registration. *Remote Sens.* **2022**, *14*, 2559. [[CrossRef](#)]
19. Wen, X.; Li, T.; Han, Z.; Liu, Y. Point Cloud Completion by Skip-Attention Network with Hierarchical Folding. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1936–1945.
20. Zong, D.; Sun, S.; Zhao, J. ASHF-Net Adaptive Sampling and Hierarchical Folding. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; pp. 3625–3632.
21. Zhang, W.; Yan, Q.; Xiao, C. Detail Preserved Point Cloud Completion via Separated Feature Aggregation. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020.
22. Zhao, Y.; Birdal, T.; Deng, H.; Tombari, F. 3D Point Capsule Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1009–1018.
23. Xia, Z.; Liu, Y.; Zhu, X.; Ma, Y.; Li, Y.; Hou, Y.; Qiao, Y. SCPNet: Semantic Scene Completion on Point Cloud. *arXiv* **2023**, arXiv:2023.06884.
24. Wei, M.; Zhu, M.; Zhang, Y.; Sun, J.; Wang, J. Cyclic Global Guiding Network for Point Cloud Completion. *Remote Sens.* **2022**, *14*, 3316. [[CrossRef](#)]
25. Huang, X.; Zhang, J.; Wu, Q.; Fan, L.; Yuan, C. A Coarse-to-Fine Algorithm for Matching and Registration in 3D Cross-Source Point Clouds. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2965–2977. [[CrossRef](#)]
26. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7659–7667.
27. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 206–215.
28. Pan, L. ECG: Edge-aware Point Cloud Completion with Graph Convolution. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4392–4398. [[CrossRef](#)]
29. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning Representations and Generative Models for 3D Point Clouds. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 40–49.
30. Pan, L.; Chen, X.; Cai, Z.; Zhang, J.; Liu, Z. Variational Relational Point Completion Network. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 8520–8529.

31. Wang, X.; Ang, M.H.; Lee, G.H. Cascaded Refinement Network for Point Cloud Completion. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 787–796.
32. Tang, J.; Gong, Z.; Yi, R.; Xie, Y.; Ma, L. LAKe-Net: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 1716–1725.
33. Wen, X. PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 7439–7448.
34. Chung, J.; Culcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
35. Lei, T.; Zhang, Y. Training RNNs as Fast as CNNs. *arXiv* **2017**, arXiv:1709.02755.
36. Mittal, H.; Okorn, B.; Jangid, A.; Held, D. Self-Supervised Point Cloud Completion via Inpainting. *arXiv* **2021**, arXiv:2111.10701.
37. 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, Honolulu, HI, USA, 27 January–1 February 2019.
38. Huang, Z.; Wang, X.; Wei, Y.; Huang, L.; Shi, H.; Liu, W.; Huang, T.S. CCNet: Criss-Cross Attention for Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
39. Charles, R.Q.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5105–5114.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.