remote sensing

MDPI

Article

Cyclic Global Guiding Network for Point Cloud Completion

Ming Wei [1,2], Ming Zhu [1,*], Yaoyuan Zhang [1,2], Jiaqi Sun [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.); zhangyaoyuan19@mails.ucas.ac.cn (Y.Z.); sunjiaqi19@mails.ucas.ac.cn (J.S.); wangjiarong@cust.edu.cn (J.W.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: zhuming@ciomp.ac.cn

Abstract: The application of 3D scenes has gradually expanded in recent years. A 3D point cloud is unreliable when it is acquired because of the performance of the sensor. Therefore, it causes difficulties in utilization. Point cloud completion can reconstruct and restore sparse and incomplete point clouds to a more realistic shape. We propose a cyclic global guiding network structure and apply it to point cloud completion tasks. While learning the local details of the whole cloud, our network structure can play a guiding role and will not ignore the overall characteristics of the whole cloud. Based on global guidance, we propose a variety of fitting planes and layered folding attention modules to strengthen the local effect. We use the relationship between the point and the plane to increase the compatibility between the network learning and the original sparse point cloud. We use the attention mechanism of the layer overlay to strengthen the local effect between the encode and decode. Therefore, point clouds are more accurate. Our experiments indicate the effectiveness of our method on the ShapeNet, KITTI, and MVP datasets and are superior to other networks.

Keywords: point cloud completion; lidar data; deep learning; point cloud processing

Citation: Wei, M.; Zhu, M.; Zhang, Y.; Sun, J.; Wang, J. Cyclic Global Guiding Network for Point Cloud Completion. *Remote Sens.* **2022**, *14*, 3316. https://doi.org/10.3390/rs14143316

Academic Editor: Sander Oude Elberink

Received: 17 June 2022
Accepted: 6 July 2022
Published: 9 July 2022

1. Introduction

Point cloud data obtained directly by radar are the most direct, effective, and convenient representation of 3D data. Each point in the point cloud represents a measured value in the 3D space, including the coordinate value x, y, and depth value z of the ordinary 2D image [1–3]. It can represent a collection of massive points in the same spatial reference system. It expresses the spatial distribution and surface features of the target. However, the sparsity of point clouds has always been a serious problem for radar perception [4,5]. How to make sparse point clouds dense and complete is the focus of research [6,7]. Scene-based completion is difficult because of the extreme uncertainty of object and position. Therefore, researchers commence with the individual object and conduct direct completion studies on the point clouds of targets. In addition, current point cloud completion studies only focus on sparse and incomplete point clouds without occlusion interference.

Through the research work GR-Net [8], 3D grids are introduced as intermediate representations to regularize unordered point clouds and to supply the loss of details. It uses a 3D convolution in the point cloud completion network as a feature extraction and has achieved good results. This leads to inadequate learning of structure and context. The regularization of disordered point clouds leads to a deviation of each point [9], which is low but not negligible in accumulation. It reduces some of the unevenness and, in some shapes, causes the overall effect to deviate [10]. However, the accuracy of the overall structure is also vital.

Sometimes, we request high requirements for detail point clouds. However, there are other times when the overall structure of a point cloud is more vital. For example, for outdoor driving, intelligent vehicles request to judge the object and position of surrounding

objects according to Lidar. At this point, the tightness and structural accuracy of the point clouds are vital for safe driving.

To solve this pivotal problem and ensure the accuracy of local details and the global structure, we implement cyclic global guidance for the whole network structure in this paper. As we all know, low resolution blurs details. However, at the same time, it costs a degree of global construction at a relatively low cost. On the one hand, we use a double-branch structure in the mesh part, which will retain a complete low-resolution global shape, and reduce the loss of resolution reduction in guiding 3D convolution continuous sampling. On the other hand, we still use the overall global structure to guide the process of gradual up-sampling. These features maintain structural features to enhance the effectiveness of the network.

In addition, Folding-Net [11] proposed a new decoding operation called folding and showed that it is theoretically universal in point cloud reconstruction. However, the 2D grid points of the folding operation are randomly generated and are usually independent of the actual point clouds. Therefore, to reflect the overall features of the point clouds, we use the multiple fitting planes to generate a 2D grid with global features. To improve the quality of the reconstructed point clouds effectively, we fold them into point cloud features between up-sampling and down-sampling. The attention mechanism can select the focusing position to produce a more discriminative feature representation. We integrate folding operation and attention module and use grid points to conduct feature analysis of point cloud information layer by layer to achieve a better point cloud completion effect. It works better through cyclic guidance of up-sampling and down-sampling.

In addition, 3D convolution and 2D convolution have similarities and universalities. With the continuous development of convolution compared with other point cloud completion methods, our network is more promising for the application. Point cloud completion methods mainly include traditional methods and methods of deep learning [12]. Traditional methods include geometry-based methods and allocation-based methods. However, they both require the use of geometric attributes and shapes of objects or complete databases. Due to the high requirements for original information, it does not apply to realistic scenes and cannot complete most of the missing point clouds. The geometric order of the points does not affect its representation of the overall shape in space because the point clouds are disorderly. Therefore, using a neural network to process point clouds directly faces serious challenges. Many basic frameworks for point cloud processing networks have emerged in recent years.

MLP-based: Point-Net [13,14] overcame the influence of the disorder and rotation of point clouds on neural network input and constructed the backbone framework of the point cloud processing network based on MLP (multilayer perceptron), which makes it possible to deal directly with point clouds and avoids the intermediate processing of original data. Since then, people have begun to study it. Folding-Net [11] proposed a novel folding-based decoder and deformed a canonical 2D grid onto the underlying 3D object surface of a point cloud. It can achieve low reconstruction errors, even for objects with delicate structures. PCN [15] operated on the original point cloud directly without any structural assumptions or comments on the underlying shapes. Through the design of the decoder, dense and complete point clouds are generated in the missing area of the input with varying degrees of incompleteness and noise to complete the realistic construction. 3D point capsule networks [16] designed an auto-encoder to deploy the dynamic routing scheme and the peculiar 2D latent space. MSN [17] merged coarse-grained prediction with input point clouds using a novel sampling algorithm. Top-Net [18] generated the structured point cloud without assuming any specific construction or topology on the underlying point set. PMP-Net [19] mimicked the behavior of an earthmover and predicted a unique point moving path for each point according to the constraint of point moving distances to improve the quality of the predicted complete shape. MLP has a poor understanding of context. It requests the addition of additional content to the network that enriches semantic information.

GAN-based: The two networks of GAN (generator and discriminator) are trained and compete in minimization maxima algorithms [20]. Achlioptas et al. [21] trained the deep Auto Encoder (AE) to learn a latent representation first and then train a generative model in that fixed latent space to distinguish synthesized from realistic samples. RL-GAN-Net [22] converted noisy partial point clouds into a high-fidelity completed shape by controlling the GAN. Based on PCN [15], PF-Net [23] estimated the missing point cloud hierarchically by utilizing a feature-points-based multi-scale generating network and used the reinforcement learning agent to control GAN to reduce the prediction time of point cloud completion. Spare-Net [24] regarded the shape feature as a style code that modulates the normalization layers during folding, which considerably enhances its capability. However, compared with the other models, GAN exists in two different networks, not in a single end-to-end network. Sometimes, the training process is erratic.

Graph-based: The point clouds inherently lack topological information. Designing a recover topology-based graph CNN (convolutional neural networks) can enrich the representation power of point clouds [25]. DGCNN [26] proposed a new neural network module, dubbed Edge-Conv, which can be stacked and applied to learn global shape properties. Edge-Conv constructed a local graph explicitly and learned the embeddings for the edges. In multi-layer systems, affinity in feature space captures semantic features over potentially long distances in the original embedding. Song et al. [27] processed the category nodes through a graph convolutional network to generate the global priors adapted to point clouds, and used them for point cloud segmentation. LDGCNN [28] eliminated the transformation network from DGCNN [26] and connected hierarchical features from different dynamic graphs to calculate informative edge vectors and avoid vanishing gradient problems. However, the feature extraction effect of the graph convolution structure may not be as good as that of convolution, and the learning ability of the point cloud may be insufficient. It is possible to increase feasibility by adding branching and deep optimization.

3D Conv-based: 3D convolution is developing in various fields related to point clouds. There will be broader room for progress in the future. 3D convolution can extract point cloud features after regularization. Point-Cnov [29] proposed a density convolution that can fully approximate 3D continuous convolution on any set of 3D points. Hua et al. [30] presented a convolutional neural network for semantic segmentation and object recognition with 3D point clouds. Lei et al. [31] proposed an octree-guided neural network architecture and spherical convolutional kernel for machine learning from arbitrary 3D point clouds. The network architecture capitalizes on the sparse nature of irregular point clouds and hierarchically coarsens the data representation with space partitioning. GR-Net [8] introduced 3D grids as intermediate representations to regularize unordered point clouds and convert between point clouds and 3D grids without losing structural information. Although the method achieves relatively accurate results, the detailed features are still lost after quantization. Therefore, it is necessary to have a good grasp of the details and overall features by learning about global and local features circularly.

Transformer-based: With the transformer becoming more widely used in natural language processing and vision [32], PT [33] and PCT [34] have applied the transformer to point clouds. It is inherently permutation invariant for processing a sequence of points; therefore, it is well suited for point cloud learning. By representing the point cloud as a set of unordered groups of points with position embeddings, PoinTr [35] converted the point cloud to a sequence of point proxies and adopted a transformer encoder–decoder architecture for point cloud completion. However, the transformer is costly in construction and poor in universality. It may be improved through lightweight research of transformers in the future.

Others: Sun et al. [36] proposed a new representation method for point clouds that learns a set of quadratic terms based on static and global reference surfaces to describe 3D shapes. However, it is not applicable to point cloud completion because the point clouds are incomplete.

In summary, frameworks for point cloud processing are branching out. However, with the improvement of 2D information processing, many innovative ideas about convolution have been put forward. 3D convolution is similar to 2D convolution, and many methods are applicable. Therefore, we believe that a framework based on 3D convolution has rosy prospects for development. In general, we designed corresponding modules specifically to improve the effect of local details and global structure and to strengthen the final completion effect.

We propose a network with cyclic global guidance for point cloud completion. Our main contributions to the paper are summarized as follows:

1. We designed the global guided down-sampling and up-sampling constructions. The complete and dense point clouds are reconstructed by combining overall construction with contextual semantic information.
2. We integrated the traditional fitting plane of point clouds adapted to point cloud features into the deep learning network, which uses the original features of point clouds to reduce uncertainty.
3. We combine a folding operation with an attention mechanism to complete the point cloud by stratification for focusing position creatively.

2. Methods

In this section, we introduce the overall structure of our CGG-Net. As shown in Figure 1, our network consists of one main route and several sub-routes. We use 3D convolution as a feature extraction framework, which refers to the idea of GR-Net [11]. The overall shape is an innovatively guided optimization that considers global influence from various aspects and perspectives.

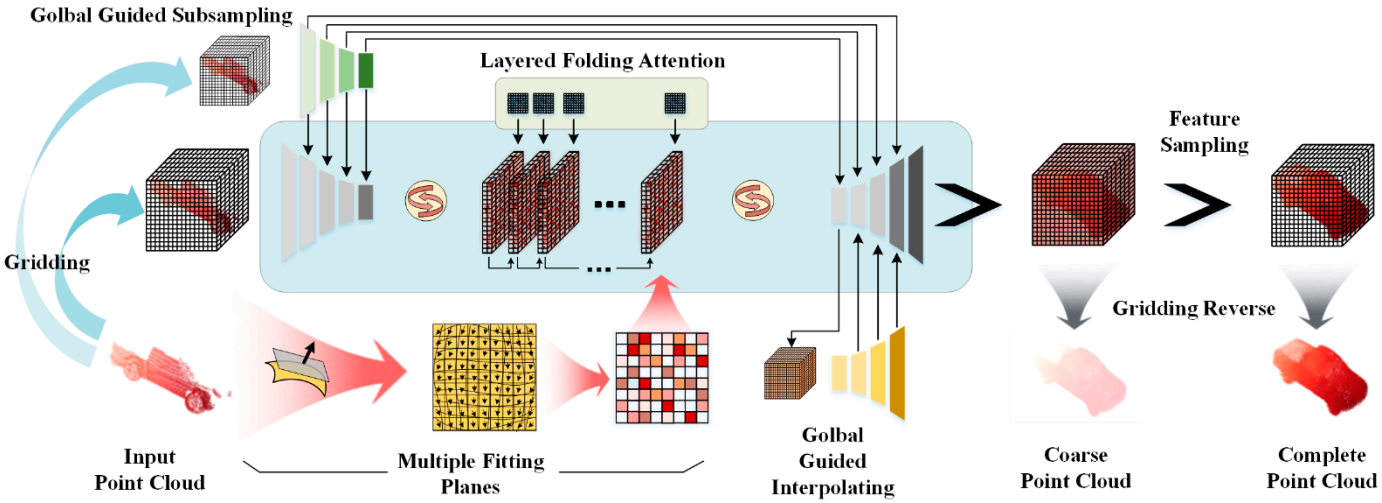

Figure 1. Construction of our network. The input point clouds are initially gridding into two scales with different resolutions. High resolution is the backbone. Low resolution acts on stacked 3D convolution for global guided sampling. The attention module is layered and folded to increase the learning effort of the network. At the same time, the multiple fitting planes are obtained from the input point clouds integrated into the results as a vital branch. We also do global bootstrapping in up-sampling. Finally, the point clouds are complete by gridding reverse and feature sampling.

From the perspective of the cascade, our circular construction network includes a global guided down-sampling module, layered folded attention module, multiple fitting plane module, and global guided up-sampling module. They are interlinked and distributed in various parts of the network construction to work together cyclically to guide the learning of the whole in the correct direction. In addition, the modules also work on the network in parallel. For the basic framework, we carried out global guided sampling to improve the learning effect of the network on the overall construction in the process of

convolution of meshed point clouds. At the same time, we always pay attention to global structural integrity in our network architecture through folding and attention modules based on overall construction.

2.1. Global Guided Sampling

The global guided sampling module is the backbone framework of our network. By considering the whole and the details of the point clouds, the network can produce better results. Zhang et al. [37] designed a feature-expansion module. It can learn local and global point features via a down-feature operator and an up-feature operator. Furthermore, we do not handle both the global and the local in one way. We propose integrating global-based guidance into two samplings. A new global guided branch connects to the main route. It plays a role in reducing the loss of the overall construction and gathering dispersed points. After each convolution, the whole will be aggregated through this branch to prevent the loss of global construction. In the process of convolution, the branch will constantly supervise and correct the learning direction of convolution and integrate the original point cloud information to obtain a compact internal representation.

In addition, the original 3D convolution sampling will result in a loss due to the scale drop. We use the parallel structure of the main and branch in convolution. The global guided sampling network can not only guarantee the effect of the global structure in the process of convolution but also reduce the loss of most resolution because of the dense fusion of the main path and the branch path. As the main network, the guided network is also multi-resolution, which can be targeted to compensate for the loss of different resolutions. We still use a global bootstrap to reduce losses once again when upsampling.

The output of the layer ω can be defined as

$$(\Gamma^{\omega})_{\frac{1}{2}\theta} = \Psi(F(\Gamma^{\omega-1})_{\theta}, F(\Gamma_{j_i}^{\omega-1})_{\theta}) \tag{1}$$

$F(\cdot)$ is the convolution. $\Gamma^{\omega-1}$ is the main output of the $\omega - 1$ layer. The global guide branch $\Gamma_{j_i}^{\omega-1} \in \Re(i = 1, \ldots, n)$ is the lead range of each branch. j is the number of branches and i is the serial number of the branches. θ is the resolution of the feature. $\Psi(\cdot)$ is the way of connecting branches. We discovered that connecting worked better than superposing. Connections can preserve more differences between branches instead of merging information, which would lose the original features. In addition, the connection parameter is for the number of channels, not the resolution size.

Hourglass 3D convolution stacked for feature extraction. There is a loss of global features. We designed an auxiliary global branch to guide sampling at each step. As depicted in Figure 2, our network gradually refines and filters shapes through the overall structure. At the same time, the resolution of our overall construction also changes to accommodate different stages of sampling. The aircraft point clouds are refined gradually by cyclic sampling. Step by step, it becomes denser and has less edge noise.

Then, we interpolate the low-resolution features back to the high resolution. The branching construction is similar to the subsampling mentioned above. Features are extracted gradually through stacking of 3D convolutions, which is necessary for details.

The output of the layer ω can be defined as

$$(\Gamma'^{\omega})_{2\theta} = \Psi(F(\Gamma'^{\omega-1})_{\theta}, F(\Gamma'^{\omega-1}_{j_i})_{\theta}) \tag{2}$$

The construction can fabricate the absence of global features in the sampling process through a cyclic global guide. The network grasps the overall direction of feature learning.

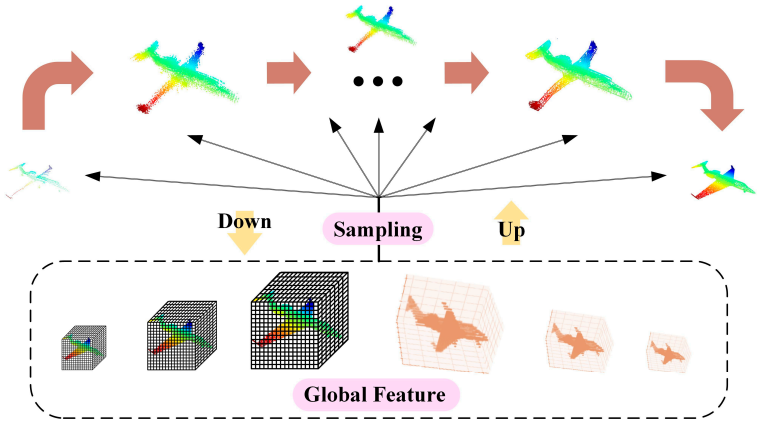

Figure 2. The structure of global guided sampling. The aircraft point clouds are restored from left to right with different resolutions by guidance. The shape is refined gradually with the noise reduced.

2.2. Multiple Fitting Planes

Folding 2D grids into incomplete points has been proven to work [10]. However, the randomly generated grid loses its connection to its construction. A multiple fitting plane module is proposed to establish this global relation. Based on the idea of the virtual point method, VR-NET [38] learns a new virtual point to form a pair of consistent point clouds. In point cloud completion, we can form corresponding virtual relationships for each or multiple point clouds. The corresponding relationship is then used as the basic parameter of point cloud completion to generate features.

Point clouds have disorder and rotation. Most feature representation methods commonly used by other data forms are not applicable to point clouds [39]. However, the normal vector, the distance to the center of gravity point, and the average distance of the neighbor node parameters are the best choices for the representative features of point cloud information. Because of this information, point cloud translation, rotation, and noise points cannot cost a lot of change. The fitting plane is the projection of the 3D point cloud to 2D and represents the traditional features of the sub-point set. We proposed multiple fitting planes to integrate the traditional point cloud representation method into a deep learning network. It can learn the overall construction of the point clouds better.

The fitting plane is precomputed before learning. For a point cloud $P = \{p_i(x_i, y_i, z_i)|i = 1, \ldots, N\}$, the nearest neighbor point of the point p_i can be defined as

$$P' = \{p_{ij}(x_{ij}, y_{ij}, z_{ij})|j = 1, \ldots, k\} \tag{3}$$

where N is the size of the point cloud, k is the number of neighbor points.

Neighborhood centers O_i can be obtained as

$$O_i = \frac{1}{k}\sum_{j=1}^{k} p_{ij} \tag{4}$$

Then, the normal vector of the point can be obtained by solving the covariance matrix of the point T_i. λ_3 is the minimum eigenvalue of the solution.

$$T_i = \begin{bmatrix} p_{i1} - O_i \\ p_{i2} - O_i \\ \cdots \\ p_{ik} - O_i \end{bmatrix}^T \begin{bmatrix} p_{i1} - O_i \\ p_{i2} - O_i \\ \cdots \\ p_{ik} - O_i \end{bmatrix} \tag{5}$$

Therefore, λ_3 is the normal vector in which the point fits into the plane. In the representation of vectors, the normal vector of the point is obtained after uniformization as

$$N(\lambda_3) = (a, b, c) \tag{6}$$

In the representation of a plane, the fitting plane of $p_i(x_i, y_i, z_i)$ is calculated as

$$C_i : a(X - x_i) + b(Y - y_i) + c(Z - z_i) = 0 \tag{7}$$

Folding-Net [11] used two consecutive three-layer perceptions to wrap a constant 2D mesh into the shape of an input point cloud. Because the network incorporates mapping from lower dimensions, it increases the possibility of more points. However, 2D meshes are always random. The randomness of point clouds for different objects is the same. Therefore, we propose that the fitting plane be used as a feature to construct the mesh to guide the folding effect. For different sparse point clouds, the constructed grid is closely related to itself and irreplaceable, which can improve the folding effect.

The set of fitting planes of each point in the point cloud $T = \{C_i | i = 1, \ldots, N\}$ can represent the features of the whole point clouds. The planes computed by different sub-points are discontinuous. The normal vectors are discrete. 2D grids are constructed by discrete normal vectors according to the quadrants of the coordinate system, with the normal vectors of the surface making the 2D features correspond to 3D. In this way, the global features can make the most of guiding the folding.

As shown in Figure 3, we have fully integrated the traditional optimal plane of the point cloud with the folding method of deep learning. This helps to improve the density and compactness of the point clouds. Learning is in close contact with the original construction. It can output better reconstruction point clouds.

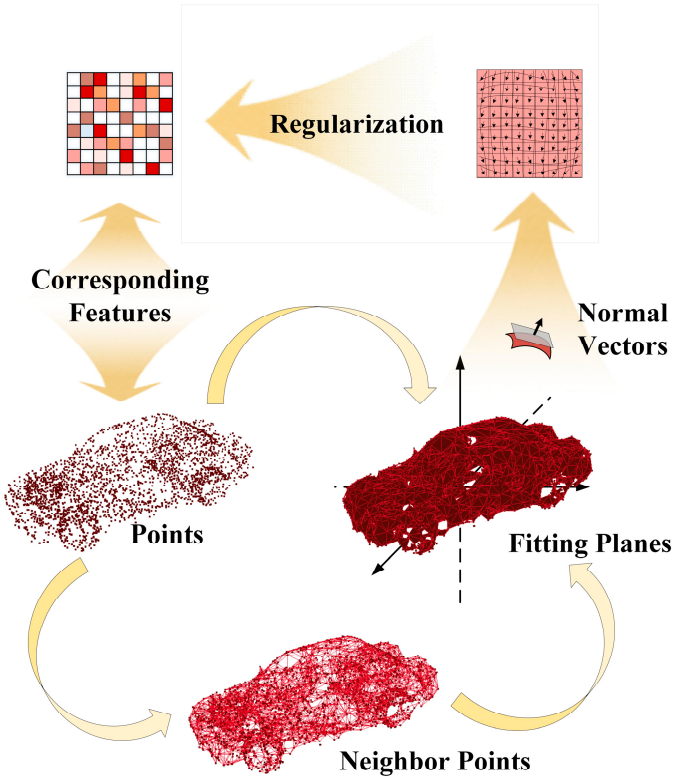

Figure 3. The construction of multiple fitting planes. We obtain neighbor points for each point of the point cloud car. We then calculated the optimal fitting plane through every sub-point. The fitting planes, calculated by clustering the subpoints in the neighborhood, are divided into eight groups according to quadrants. We receive globally targeted data features and finally map them to the initial values of a two-dimensional grid for the subsequent step of folding.

2.3. Layered Folding Attention

We still pay special attention to global construction in the layered folding attention module. We integrated the attention mechanism into the network in layers. Every layer must be combined with the last result to ensure the accuracy of the learning direction.

Zhang et al. [40] proposed a network combining multi-scale hierarchical feature fusion to gradually and adaptively improve network performance in image defogging. The layered approach can reduce feature redundancy and learn from compact internal representations. The attention mechanism is widely used in neural network architecture. Its essence is to assign different attention to the input weights to maximize context and better results. However, it operates on the features themselves or on residual constructions [41]. We put forward the idea of adding folding points to the network through attention in Figure 1. The attention is no longer limited to its feature extraction but becomes a better way to fusion and obtain more effective information. We designed a layered attention mechanism and used it for further context fusion of features and enhanced information association layer by layer.

The outlier noise point of the point cloud is also a problem for the point cloud completion task. We believe that outlier noise points are caused by insufficient context processing of the local details of point clouds. Therefore, our layered folding attention module is used in the intermediate step with feature extraction. It can perform contextual fusion and processing of features in convolution layer by layer, which can eliminate deviation points and error points.

As shown in Figure 1, point cloud features are layered first. Then, they are fused layer by layer in our network, starting with the features of the grid and the top layer and working down. We integrate the grid into the network rather than fold it directly.

The output Ω_n can be regarded as follows:

$$\begin{cases} \Omega_n = \Psi(F(G), \Omega_{n-1}) & n = 1 \\ \Omega_n = \Psi(P_{n-1}, F(\Omega_{n-1})) & n > 2 \end{cases} \tag{8}$$

where n is the number of layers, G is the grid to be folded, $F(\cdot)$ is the variable convolution, P is the slice of the input feature, and $\Psi(\cdot)$ is a connection method of computing.

2.4. Others

Our network references the GR-Net [8] backbone framework. 3D convolution is applied to feature extraction of the point cloud by gridding and gridding reverse. We obtained the final point clouds after feature sampling. In this section, we introduce the basic modules from GR-Net [8].

2.4.1. Gridding and Gridding Reverse

To preserve the spatial layout and further processing, the point cloud $p_i(x_i, y_i, z_i)$, $i = 1, \ldots, n$ with n points is represented as a regular 3D grid T with N resolution consisting of two parts V and W.

$$T = \langle V, W \rangle \tag{9}$$

where $V = \{v_i\}, i = 1, \ldots, n$ represents eight neighbor coordinates vertices in each cell. $W = \{w_i\}, i = 1, \ldots, n$ is the point of the cell.

v_i is defined as:

$$v_i \in \left\{ (-\frac{N}{2}, -\frac{N}{2}, -\frac{N}{2}), \ldots, (\frac{N}{2} - 1, \frac{N}{2} - 1, \frac{N}{2} - 1) \right\} \tag{10}$$

According to GR-Net [8], w_i is defined as

$$\begin{cases} w_i = \sum\limits_{p \in N(v_i)} \frac{\omega(v_i, p)}{|N(v_i)|} & |N(v_i)| \neq 0 \\ w_i = 0 & |N(v_i)| = 0 \end{cases} \tag{11}$$

$$\omega(v_i, p) = (1 - |x_i - x|)(1 - |y_i - y|)(1 - |z_i - z|) \tag{12}$$

where $|N(v_i)|$ is the number of neighboring points of v_i.

Conversely, a grid inverse operation is required to recover each of the eight neighborhood points $\Theta_i, i = 1, \ldots, 8$ of each cell grid at the end of the network, which is defined as

$$\begin{cases} p_i^c = \frac{\sum w_\theta{}' v_\theta}{\sum w_\theta{}'} & \sum w_\theta{}' \neq 0 \\ ignore & \sum w_\theta{}' = 0 \end{cases}, \theta \in \Theta_i \tag{13}$$

$w_\theta{}'$ and v_θ still represent the coordinates and the values of the eight vertices.

2.4.2. Feature Sampling

Max-pooling may cause the loss of local context information and affect the recovery effect of details. Therefore, feature sampling replaces it according to GR-Net [8]. We connected eight neighbor points in the cell grid in series as a feature of this point. Points are extracted from the coarse point cloud to form a complete point cloud.

2.5. Loss Function

We take the Chamfer Distance (CD) as the loss function [2]. For point clouds S_1 and S_2, the CD is defined as

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2 \tag{14}$$

CD measures the average distance between each point in a point and the nearest point in another point clouds [42]. It is also the most extensive way to judge the completion effect.

3. Results and Discussion

We implemented the network using Pytorch and used the Adam optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.999$ during training. The initial learning rate is set to 10^{-4} and is decayed by 0.1 at 100 and 150 epochs. We train the network for 200 epochs with 32 batch sizes on one GeForce RTX 2080 Ti GPU. It takes about 8 min per epoch on ShapeNet.

We use the CD and the F-Score (FS) to measure the effectiveness of the network [43]. The CD is defined in the loss function section and is more computationally efficient and widely used. The *FS* is defined as follows:

$$FS = \frac{2P(d)R(d)}{P(d) + R(d)} \tag{15}$$

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.

We use Uniformity from PU-GAN [20] to test the integrity of the point clouds. This can better show the effect of the global completion we pursue. Uniformity includes global and local measures. It is defined as follows:

$$Uniformity(P) = \frac{1}{N} \sum_{i=1}^{n} U_{imbalance}(S_i) U_{clutter}(S_i) \tag{16}$$

$$U_{imbalance}(S_i) = \frac{(|S_i| - \hat{n})^2}{\hat{n}} \tag{17}$$

$$U_{clutter}(S_i) = \frac{1}{|S_i|} \sum_{j=1}^{|S_i|} \frac{(d_{i,j} - \hat{d})^2}{\hat{d}} \tag{18}$$

where S_i is a subset of point clouds P. It was calculated using the farthest sampling and the ball query of radius. And $\hat{n}$ is the expected number of points in S_i, $d_{i,j}$ is the distance to the nearest neighbor for the $j-th$ point.

If S_i has a uniform distribution, $\hat{d}$ is defined as

$$\hat{d} = \sqrt{\frac{2\pi P}{\left|S_i\right|\sqrt{3}}} \tag{19}$$

3.1. Results of Comparative Experiments

3.1.1. ShapeNet

The ShapeNet from PCN [14] includes eight categories, such as aircraft, automobiles, etc. There are 30,974 models of ground truth extracted from the categories. This makes the input distribution more similar to the sensor data in the real world. The 16,384 sampling points of the object were back-projected to 3D through 2.5D depth images to generate partial point clouds rather than a subset of complete point clouds [44]. The number of sparse point clouds for each object is not fixed. Most of them are in the range of 500 to 2000. Through a deep learning network, we complete the number of a point cloud of the object to 16,384 points and compare them with the ground truth.

A comparison of our network effects is shown in Table 1. The CD of most categories is significantly lower than that of other networks. The overall CD of our network is approximately 0.04 less than that of GR-Net [8] (using the same loss function and the same training parameters). At the same time, our CD results are much better than those of other networks.

Table 1. The results of the experiment using chamfer distance $\times 10^3$ with L2 norm on ShapeNet (lower is better). The bold numbers are the best.

Model	F-Net	Top-Net	MSN	GR-Net	Spare-Net	CGG-Net
Airplane	0.62	0.22	0.25	0.29	**0.18**	0.27
Cabinet	1.61	0.56	0.97	0.63	0.66	**0.59**
Car	0.62	0.35	0.45	0.32	0.36	**0.31**
Chair	1.55	0.63	0.77	0.55	0.62	**0.54**
Lamp	2.03	0.75	0.93	0.58	0.63	**0.38**
Sofa	1.54	0.69	1.15	**0.69**	0.79	0.76
Table	1.53	0.48	0.67	0.48	0.50	**0.41**
Watercraft	0.91	0.44	0.49	0.31	0.38	**0.29**
Overall	1.30	0.52	0.71	0.48	0.52	**0.44**

Our point cloud completion results on ShapeNet [24] are shown in Figure 4. Our CGG-Net completes the missing shape well and makes it close to the ground truth with less noise.

The time of the network is also vital. It can ensure real-time performance and wide application. As shown in Table 2, our network has the best CD value. Although Spare-Net [24] has the best FS value, it uses GAN construction. In addition, although the FS of our network is only about 0.035 less than Spare-Net [24], the time is almost 100 times better. Although our network time is only about 0.002s longer than GR-Net [8], the FS improves by about 0.01. This proves that our network has an advantage in effect and time.

Table 2. Results of comparative experiments on ShapeNet. The bold numbers are the best.

Model	GAN	CD	FS	Time
Spare-Net	√	0.52	**0.6607**	1.055
GR-Net	×	0.48	0.6179	**0.020**
CGG-Net	×	**0.44**	0.6266	0.022

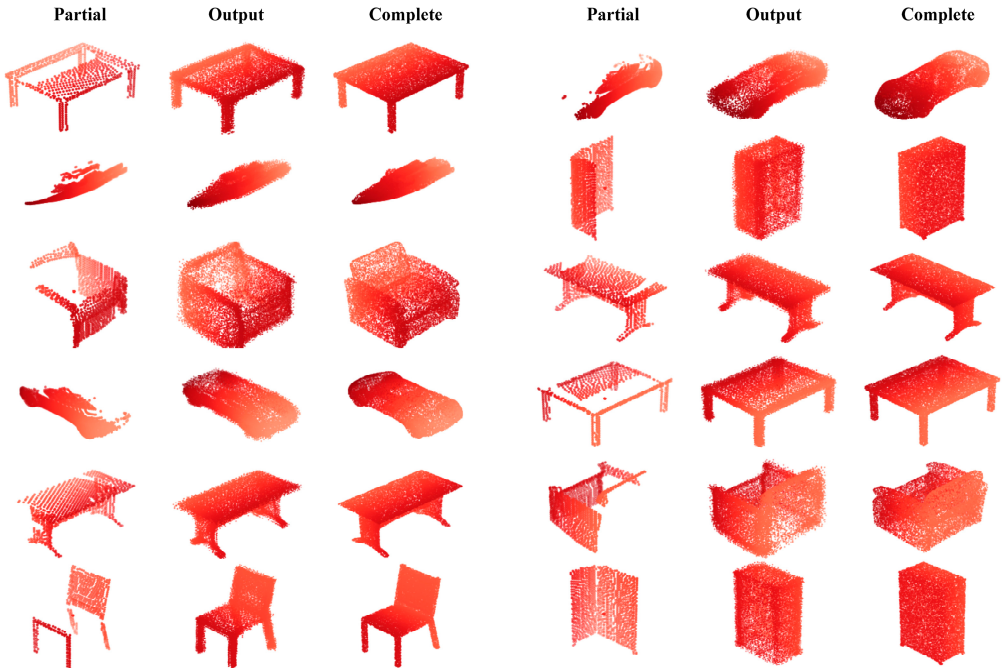

Figure 4. The result of point cloud completion on ShapeNet. The partial and missing point clouds are in the left column. The ground truth value is in the right column. The completion point clouds we output are in the middle column.

3.1.2. KITTI

KITTI without ground truth is also from PCN [15]. PCN [15] extracts the point clouds according to the points in the object surrounding the box of the car, which generates 2483 partial point clouds. Finally, each point cloud is converted to a world coordinate system [45]. Each car has a smaller number of points, with most having fewer than 500 points. For unity, we still reconstruct it to 16,384 points.

We show the completion effect in Figure 5. The shape of the car is completed nicely through our network. By training the vehicle model on ShapeNet, the effective network parameters for point clouds are obtained to reconstruct KITTI. As shown in Table 3, we use Uniformity to measure the effect of point clouds. Our network is significantly better at recreating the car. In all the percentages calculated, our vehicle construction as a whole receives the best completion.

Table 3. The uniformity results of comparative experiments on ShapeNet-car and KITTI (lower is better). The bold numbers are the best.

Datasets	Percentage	Atlas-Net	PCN	F-Net	Top-Net	MSN	GR-Net	CGG-Net
ShapeNet-Car	0.6%	-	-	-	-	-	0.23	**0.12**
	0.8%	-	-	-	-	-	0.14	**0.11**
	1.0%	-	-	-	-	-	0.08	**0.07**
	1.2%	-	-	-	-	-	0.05	**0.04**
KITTI	0.6%	1.01	5.81	1.30	1.32	0.68	0.27	**0.27**
	0.8%	0.87	7.71	1.26	1.22	0.52	0.20	**0.19**
	1.0%	0.76	9.33	1.16	1.07	0.46	0.15	**0.13**
	1.2%	0.69	10.82	1.06	0.95	0.38	0.12	**0.10**

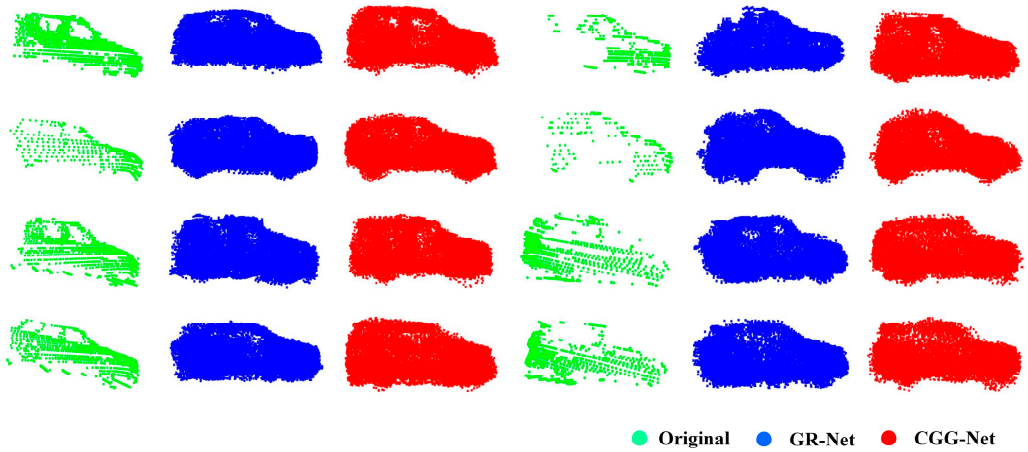

Figure 5. The result of point cloud completion on KITTI. The point clouds are missing at different angles due to the performance of the lidar and noise. The green realistic world original point clouds are very sparse and uncertain. The point clouds are missing at different angles due to the angle and performance of the lidar. The results of GR-Net [11] are shown in blue. The results of the CGG-Net are in red. As shown, the red car silhouette is more realistic. We can clearly obtain the position of the wheels with less noise.

3.1.3. MVP

The MVP is a high-quality, multi-view partial point cloud dataset [6]. Besides the conventional 8 categories in existing datasets on ShapeNet, MVP allows the evaluation of 8 additional categories. It consists of over 100,000 high-quality incomplete and complete point clouds. The number of sparse point clouds is about 1000 points. Similarly, we completed 16,384 points for comparison.

Top-Net [18] selection tree-shaped mlp focuses on the overall structure. GR-Net [8] uses the same 3D convolution structure as our network. Therefore, we chose them for comparison with our network experiment on MVP to display global performance. Our results on the MVP using CD $\times$ 102 and FS are shown in Table 4. The category of MVP is similar to ShapeNet. Therefore, we chose the overall comparison results. Our network is built based on 3D convolution, so we compared it with GR-Net [8] based on the same foundation. In addition, we chose two other MLP-based networks for the experiment. CD_t and CD_p are two different CD calculation methods. Even with a little more time, our network achieved better results.

$$CD\_t = mean(CD_1) + mean(CD_2) \tag{20}$$

$$CD\_p = 0.5(mean(\sqrt{CD_1}) + mean(\sqrt{CD_2})) \tag{21}$$

where CD_1 is the chamfering distance from the completion point cloud to the ground truth. In contrast, CD_2 is the chamfering distance from the ground truth to the completion point cloud.

Table 4. Results of comparative experiments on MVP. The bold numbers are the best.

	Model	CD_t	CD_p	FS
MLP tree-based	TOP-Net	1.915	0.120	0.299
3D Conv-based	GR-Net	1.871	0.172	0.377
	CGG-Net	**1.718**	**0.143**	**0.398**

3.2. Results of Ablation Experiments

3.2.1. Global Guided Sampling

As shown in Table 5, the CD is about 0.03 lower after global guided sampling. The FS is about 0.01 higher than before. Overall, more categories receive better results using our module.

Table 5. Ablation experiment of global guided sampling. The bold numbers are the best.

Categories/Model		Air-Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Water-Craft	Overall
CD	None	0.28	0.60	**0.33**	**0.49**	0.51	0.98	0.45	0.32	0.50
(Lower is better)	Add	**0.27**	**0.58**	0.35	0.56	**0.41**	**0.84**	**0.43**	**0.30**	**0.47**
FS	None	0.77	0.51	0.60	0.58	0.69	0.48	0.64	0.69	0.61
(Higher is better)	Add	**0.78**	**0.51**	**0.60**	**0.58**	**0.69**	**0.49**	**0.65**	**0.70**	**0.62**

It is vital to control the overall shape of the point clouds continuously. As shown in Figures 6 and 7, the point cloud completion effect is improved significantly through the global sampling module. The wing and fuselage sections of the aircraft have better shape completion and structural details, as shown in Figure 6. In Figure 7, global guided sampling makes the overall surface of the lamp smooth. The shape of the lamp cap was complete. At the same time, the surface of the lamp post becomes smooth after using the module because of the reduction of noise.

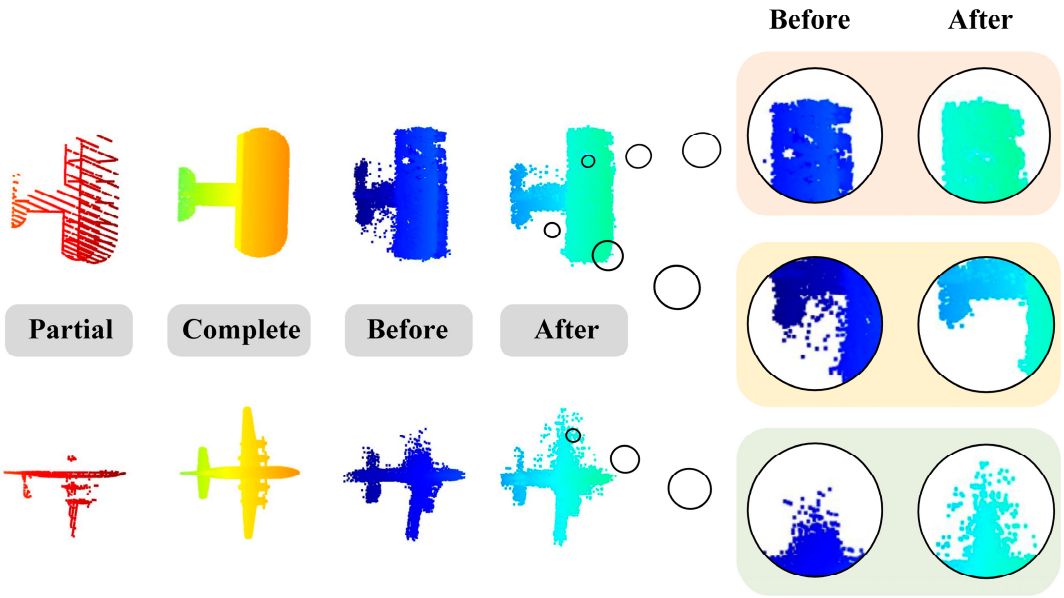

Figure 6. Global guided sampling completes the detailed shape of the aircraft. The point clouds of different colors in the two rows and four columns on the left of the picture are partial, ground truth, the complete point clouds without any processing, and complete point clouds using our module. The details of the point clouds are on the right.

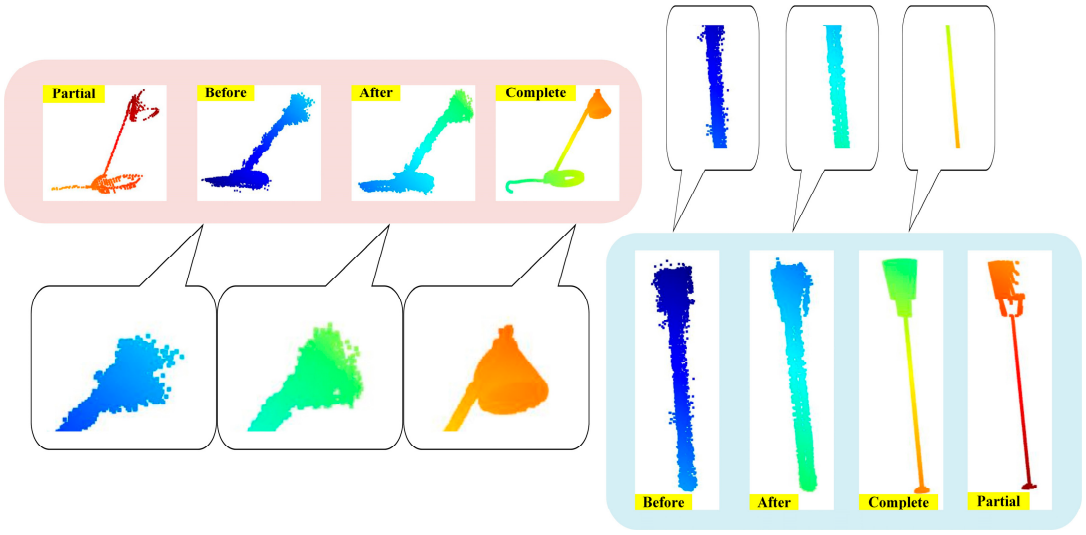

Figure 7. Global guided sampling completes the detailed shape of the lamp.

3.2.2. Multiple Fitting Planes

As shown in Table 6, the multiple fitting planes module has a significant impact on reducing CD. We reduce more than 0.04 in CD after adding the module. The advantages of the modules are also shown in Figure 8. The point clouds are denser than before with the use of modules. At the same time, the shape and contour of the point cloud object are better completed.

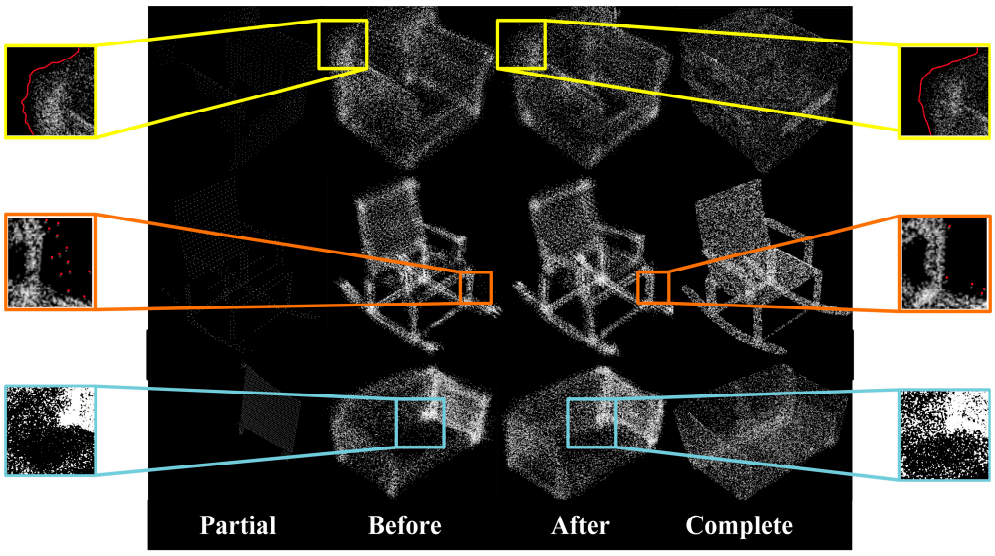

Figure 8. The effects of multiple fitting planes module. In the first line, we use the red curve to draw the shape of the point clouds after completion. The sofa has a clearer profile and is closer to the ground truth. In the second line, we used red dots to indicate the points that are not part of the point cloud shape. Our module effectively reduces outlier noise points. In the last line, we enhanced the contrast of the framed details.

Table 6. Ablation experiment of multiple fitting planes. The bold numbers are the best.

Categories	CD (Lower Is Better)	
	None	**Add**
Airplane	**0.27**	0.29
Cabinet	**0.58**	0.63
Car	0.35	**0.32**
Chair	0.56	**0.53**
Lamp	0.42	**0.42**
Sofa	0.84	**0.65**
Table	**0.43**	0.46
Watercraft	0.30	**0.30**
Overall	0.47	**0.46**

3.2.3. Layered Folding Attention

We conduct grouping experiments with different parameters for layered folding attention modules. The experimental results are shown in Table 7. The best effect can be obtained using a 2D grid with a length of 32 sides. The CD is more than 0.05 lower than before. The FS is about 0.02 higher. Figure 9 shows the point cloud completion effect visually after layered folding.

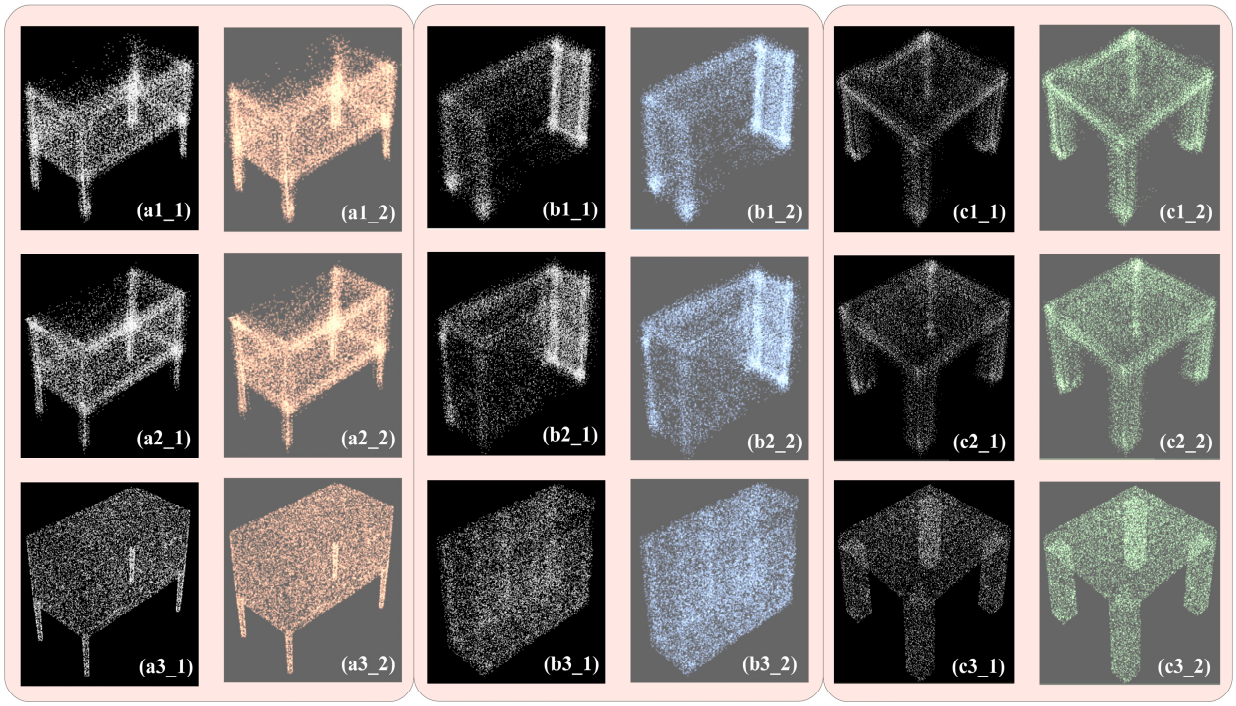

Figure 9. The effects using layered folding attention. a, b, and c are three groups of different clouds. The first line (X1_X) is the point cloud completion result without adding modules, the second line (X2_X) is the point cloud completion result that adds our attention to layering folding, and the third line (X3_X) is the ground truth value. (XX_2) is the result of discoloration and contrast stretching from (XX_1). It is better to display the contrast results. Our module works well with denseness and restores shape effectively.

Table 7. Ablation experiment of layered folding attention. The bold numbers are the best.

Categories/Model		Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Overall
CD (Lower is better)	None	0.29	0.63	0.32	0.53	0.42	**0.65**	0.46	0.30	0.46
	16	0.29	0.56	0.35	**0.51**	0.41	0.78	0.48	0.29	0.46
	32	0.27	0.59	**0.31**	0.54	**0.38**	0.76	**0.41**	**0.29**	**0.44**
	64	**0.24**	**0.55**	0.32	0.55	0.39	0.78	0.57	0.30	0.46
FS (Higher is better)	None	0.78	0.52	0.60	0.59	0.70	0.49	0.65	0.70	0.62
	16	0.77	0.51	0.60	0.58	0.69	0.48	0.64	0.69	0.62
	32	**0.79**	**0.53**	**0.61**	**0.59**	**0.70**	**0.50**	**0.66**	**0.70**	**0.64**
	64	0.78	0.52	0.60	0.58	0.70	0.49	0.65	0.69	0.63

3.2.4. Summary of the Experiment

The results of our progressive ablation experiment are shown in Table 8. Our final selections are displayed in bold. The weights and times of our network only slightly increase. However, the CD and FS are gradually better than in the previous step. At the same time, we found that the size of the grid dot was not the larger the better. For this, we think that the folded grid needs to maintain a certain kind of fitting and the suitability of the point cloud object.

Table 8. Ablation experiment of cyclic global guiding network. The bold numbers are the best.

Model	CD	FS	Weights (M)	Time (s)
Original	0.50	0.61	306.8	0.019
Original+GGS	0.47	0.62	307.0	0.020
Original+GGS+MFP	0.46	0.62	321.5	0.020
Original+GGS+MFP+LFA-16	0.46	0.62	338.3	0.022
Original+GGS+MFP+LFA-32	**0.44**	**0.64**	**338.3**	0.022
Original+GGS+MFP+LFA-64	0.46	0.63	338.3	0.022

The CD of the test dataset varies with the number of training epochs as shown in Figure 10. We took a multiple of 25 from 25 to 200 in the experiment. All of our ideas have had beneficial effects. Our CGG-Net training curve is gentler and more difficult to overfit. The expression ability is better than that of others.

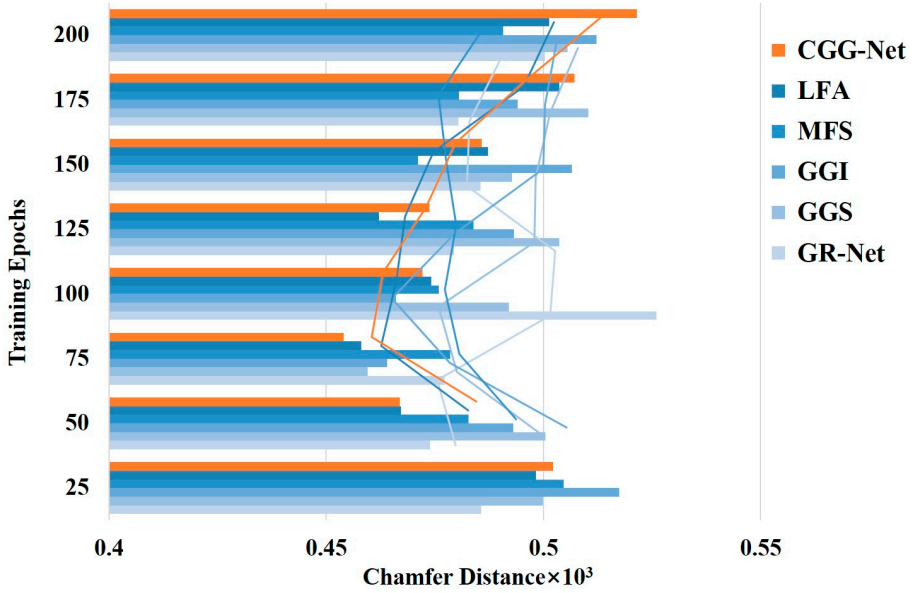

Figure 10. The learning curve of the networks. The trend of the curve is the moving average. GGS represents global guided down-sampling, GGI represents global guided up-sampling, MFS represents multiple fitting planes, and LFA represents layered folding attention.

4. Conclusions

In this paper, we propose a new CGG-Net for sparse and partial point cloud completion that pays more attention to the control of the whole construction completion. In network construction, the whole construction of the point clouds is effectively taken into account through cyclic global guidance. In addition, we propose a folding method of multiple fitting planes to integrate traditional feature representations into deep learning. Finally, the point cloud is carefully completed step-by-step through layered folding attention. We proved the effectiveness of the network in the experiment. The superiority of the network is shown by comparison with other methods on ShapeNet, KITTI, and MVP.

Author Contributions: M.W. and Y.Z. contributed to the theory research and the experiments conception and analyzed the results; M.W. and J.S. performed the experiments; M.W., Y.Z., and J.S. 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: Not applicable.

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

References

1. 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]
2. 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.
3. 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]
4. Fan, H.; Su, H.; Guibas, L. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2463–2471.
5. Mittal, H.; Okorn, B.; Jangid, A.; Held, D. Self-Supervised Point Cloud Completion via Inpainting. *arXiv* **2021**, arXiv:2111.10701.
6. 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.
7. 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.
8. 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, Glasgow, UK, 23–28 August 2020.
9. 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.
10. 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.
11. 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.
12. Pan, L. ECG: Edge-aware Point Cloud Completion with Graph Convolution. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4392–4398. [CrossRef]
13. 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.
14. Charles, R.Q.; 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.
15. 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.
16. 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.

17. 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.
18. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, 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.
19. 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, 20–25 June 2021; pp. 7439–7448.
20. Li, R.; Li, X.; Fu, C.W.; Cohen-Or, D.; Heng, P.A. PU-GAN: A Point Cloud Upsampling Adversarial Network. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 20 October–2 November 2019; pp. 7202–7211.
21. 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.
22. Sarmad, M.; Lee, H.J.; Kim, Y.M. RL-GAN-Net: A Reinforcement Learning Agent Controlled GAN Network for Real-Time Point Cloud Shape Completion. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5891–5900.
23. 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.
24. Xie, C.; Wang, C.; Zhang, B.; Yang, H.; Chen, D.; Wen, F. Style-based Point Generator with Adversarial Rendering for Point Cloud Completion. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4617–4626.
25. Velikovi, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903.
26. 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]
27. Song, Z.; Zhao, L.; Zhou, J. Learning Hybrid Semantic Affinity for Point Cloud Segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4599–4612. [CrossRef]
28. Zhang, K.; Hao, M.; Wang, J.; Silva, C.D.; Fu, C. Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features. *arXiv* **1904**, arXiv:1904.10014.
29. Wu, W.; Qi, Z.; Fuxin, L. PointConv: Deep Convolutional Networks on 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
30. Hua, B.; Tran, M.; Yeung, S. Pointwise Convolutional Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 984–993.
31. Lei, H.; Akhtar, N.; Mian, A. Octree Guided CNN with Spherical Kernels for 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9623–9632.
32. Liu, S.; Fu, K.; Wang, M.; Song, Z. Group-in-Group Relation-Based Transformer for 3D Point Cloud Learning. *Remote Sens.* **2022**, *14*, 1563. [CrossRef]
33. Engel, N.; Belagiannis, V.; Dietmayer, K. Point Transformer. *arXiv* **2012**, arXiv:2012.09164. [CrossRef]
34. Guo, M.H.; Cai, J.X.; Liu, Z.N.; Mu, T.J.; Martin, R. PCT: Point cloud transformer. *Comp. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
35. Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers. *arXiv* **2021**, arXiv:2108.08839.
36. Sun, T.; Liu, G.; Li, R.; Liu, S.; Zhu, S.; Zeng, B. Quadratic Terms based Point-to-Surface 3D Representation for Deep Learning of Point Cloud. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2705–2718. [CrossRef]
37. Zhang, P.; Wang, X.; Ma, L.; Wang, S.; Kwong, S.; Jiang, J. Progressive Point Cloud Upsampling via Differentiable Rendering. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4673–4685. [CrossRef]
38. Zhang, Z.; Sun, J.; Dai, Y.; Fan, B.; He, M. VRNet: Learning the Rectified Virtual Corresponding Points for 3D Point Cloud Registration. *arXiv* **2022**, arXiv:2203.13241. [CrossRef]
39. Miao, Y.; Zhang, L.; Liu, J.; Wang, J.; Liu, F. An End-to-End Shape-Preserving Point Completion Network. *IEEE Comput. Graph. Appl.* **2021**, *41*, 20–33. [CrossRef]
40. Zhang, X.; Wang, J.; Wang, T.; Jiang, R. Hierarchical Feature Fusion with Mixed Convolution Attention for Single Image Dehazing. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 510–522. [CrossRef]
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Shi, J.; Xu, L.; Heng, L.; Shen, S. Graph-Guided Deformation for Point Cloud Completion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7081–7088. [CrossRef]
43. Tatarchenko, M.S.; Richter, R.; Ranftl, R.; Li, Z.; Koltun, V.; Brox, T. What Do Single-View 3D Reconstruction Networks Learn? In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3400–3409.

44. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. AtlasNet: A Papier-Mché Approach to Learning 3D Surface Generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
45. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]