



# Article FuNet: Multi-Feature Fusion for Point Cloud Completion Network

Keming Li<sup>1</sup>, Weiren Zhao<sup>1,\*</sup>, Junjie Liu<sup>1</sup>, Jiahui Wang<sup>2,\*</sup>, Hui Zhang<sup>1</sup> and Huan Jiang<sup>1</sup>

- <sup>1</sup> School of Physics and Optoelectronic Engineering, Guangdong University of Technology, Guangzhou 510006, China; 2112115095@mail2.gdut.edu.cn (K.L.); 2112115025@mail2.gdut.edu.cn (J.L.); yzz2019@gdut.edu.cn (H.Z.); jianghuan@gdut.edu.cn (H.J.)
- <sup>2</sup> School of Physics, Sun Yat-sen University, Guangzhou 510275, China
- \* Correspondence: zwren@gdut.edu.cn (W.Z.); wangjh@mail.sysu.edu.cn (J.W.)

**Abstract:** The densification of a point cloud is a crucial challenge in visual applications, particularly when estimating a complete and dense point cloud from a local and incomplete one. This paper introduces a point cloud completion network named FuNet to address this issue. Current point cloud completion networks adopt various methodologies, including point-based processing and convolution-based processing. Unlike traditional shape completion approaches, FuNet combines point-based processing and convolution-based processing to extract their features, and fuses them through an attention module to generate a complete point cloud from 1024 points to 16,384 points. The experimental results show that when comparing the optimal completion networks, FuNet decreases the CD by 5.17% and increases the F-score by 4.75% on the ShapeNet dataset. In addition, FuNet achieves better results in most categories on a small sample dataset.

Keywords: point cloud; completion; feature fusion; attention module

# 1. Introduction

Point cloud, as the most common format of 3D model expression, has been widely used in computer vision [1], robotics [2], and other fields. Point cloud is playing an important role in tasks such as 3D target classification, 3D scene segmentation, and 3D reconstruction because of its simple data structure and expressive ability. However, the point cloud acquired from the real objects is often sparse and incomplete, and cannot be directly applied to some downstream tasks. Therefore, recovering a local and incomplete point cloud into a complete and dense one is crucial for practical applications.

Point cloud completion networks usually consist of an encoder–decoder structure. The encoder is responsible for extracting the point cloud feature and the decoder is responsible for generating a complete point cloud from a coarse one.

There are usually two types of methods for point cloud feature extraction: point-based processing and convolution-based processing. Point-based processing usually utilizes MLP (Multi-Layer Perceptron) to process each point independently. As the originator of point-based processing, PointNet [3] applies shared MLPs and maximum pooling operations to obtain the features of the point cloud; however, it struggles to capture the local features since the maximum pooling layer is applied to all points in the point cloud. PointNet++ [4] builds on PointNet by adding a hierarchical structure that obtains information about the geometric structure of the point cloud. Several methods attempt to project the point cloud onto regular structures to use convolutional processing. Li et al. [5] designed an X-conv operator, which implements the aggregation of neighboring point features to the centroid using MLP and convolutional operations. Wang et al. [6] propose the EdgeConv, which aims at enhancing the capture of local geometric features within the point cloud while still maintaining permutation invariance. Xu et al. [7] constructed convolutional kernels by dynamically assembling basic weight matrices stored in a weight library, and these



Citation: Li, K.; Zhao, W.; Liu, J.; Wang, J.; Zhang, H.; Jiang, H. FuNet: Multi-Feature Fusion for Point Cloud Completion Network. *Electronics* 2024, *13*, 1155. https://doi.org/10.3390/ electronics13061155

Academic Editor: Claus Pahl

Received: 31 January 2024 Revised: 16 March 2024 Accepted: 19 March 2024 Published: 21 March 2024



**Copyright:** © 2024 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/). coefficients are adaptively learned from the point locations using ScoreNet. While in the realm of point cloud completion, a prevalent trend among convolution-based methods involves the gridding or voxelizing of the point cloud before applying 3D convolution. Xie et al. [8] utilized 3D grids as an intermediate representation to handle irregular point clouds. Wang et al. [9] designed a voxel-based network that integrates the object structure information into shape completion using edge generation.

There are several methods used to generate point cloud. Yang et al. [10] implemented two folding operations to transform a fixed 2D grid into the shape of the input point cloud. Folding-based methods such as MSN [11] and PoinTr [12] typically sample 2D grids from a fixed-size 2D plane and subsequently connect them to a global shape representation extracted from a point cloud feature encoder. Yuan et al. [13] proposed a coarse-to-fine point cloud generator that combines the advantages of both the fully-connected operation [14] and the folding-based operation [10]. Wang et al. [15] used two region convolutions to convert the region features into the point cloud.

Among the point cloud completion networks developed in recent years, the vast majority of networks are implemented using point-based methods (MSN [11], PCN [13], FoldingNet [10]) or convolution-based methods (GRNet [8], SoftPoolNet [15]). These networks usually only consider one processing method, but we chose to integrate the two methods to achieve the superposition of the advantages of the two methods in this paper. In addition, some networks use a GAN-based architecture (PF-Net [16], ShapeInversion [17]), which generally can only generate a small number of points, 1024 or 2048, due to the complexity of point distribution and training. Although the point cloud generated by networks based on the Transformer architecture (PoinTr [12], SnowflakeNet [18]) is better, the number of Transformer parameters is large and the mechanism is difficult to explain.

In this paper, a novel point cloud completion network named FuNet is proposed, which combines point-based processing and convolution-based processing to extract point cloud features. And the attention module is designed to fuse the features of the two processes. The experimental results show that FuNet achieves excellent performance in point cloud completion. For example, on the ShapeNet dataset [19] used for point cloud completion, FuNet attains a CD (Chamfer Distance) [20] of 9.91 and an F-score [21] of 66.1%, which are superior to those in previous networks.

### 2. Point Cloud Completion Network

The overall framework of FuNet is shown in Figure 1; it is an encoder–decoder architecture network. The feature  $f_{pb}$  is extracted by point-based processing and the feature  $f_{cb}$  is extracted by convolution-based processing, and then two coarse point clouds,  $P_{pb}$  and  $P_{cb}$ , corresponding to them are generated. Then, the decoder fuses the above two features in the attention module to obtain the global feature  $f_G$ , which is used to generate a complete point cloud  $P_{complete}$ . The different point clouds notations are shown in Table 1.

Notation Point Cloud	
$P_{pb}$	the coarse point cloud generated by $f_{pb}$
$\dot{P_{cb}}$	the coarse point cloud generated by $\dot{f}_{cb}$
$P_{complete}$	the complete output point cloud
$P_{gt}$	the ground truth point cloud

Table 1. Notation for different point clouds.

The loss function  $\mathcal{L}$  is evaluated using both the ground truth point cloud  $P_{gt}$  and either the coarse or complete point cloud, and it is employed to train the whole network through backpropagation.





**Figure 1.** FuNet's architecture. The encoder extracts features  $f_{pb}$  and  $f_{cb}$  from the input point cloud. The decoder fuses the two features and outputs a complete point cloud.

#### 2.1. Encoder

Input

MIP

The encoder separately extracts local structure information from the point cloud by point-based processing, and global contour information by convolution-based processing.

Point-based processing. As a simple and effective network used for point cloud shape classification and part segmentation, Point-PN [22] enables point cloud feature extraction by using a series of nonparametric components and linear layers, then stacking them into multiple stages to build a pyramid hierarchy. Therefore, the extended version of Point-PN, which is designed in this paper, inherits the original structure and extract features used for point cloud completion.

Firstly, the dimensions of the input point cloud are extended by a shared MLP, which is then input into a multi-stage hierarchy. The multi-stage hierarchy applies Farthest Point Sampling (FPS), *k*-Nearest Neighbors (*k*-NN), trigonometric functions and pooling operations to progressively aggregate the local geometric structure to generate a high-dimensional feature  $f_{pb}$  representing the feature obtained from point-based processing.

At each stage of the multi-stage hierarchy, an M-points input point cloud is denoted as  $P = \{p_i\}_{i=1}^M$ , where  $p_i \in \mathbb{R}^{1 \times 3}$  represents the coordinates of a point. The number of points is downsampled from M to  $\frac{M}{2}$  by FPS. Then, k-NN are responsible for dividing k neighborhoods from M points for each center c to form a local 3D region, and the value of k is 8 in our network. Normally, the combination of FPS and k-NN is used to extract the set of local neighborhood points and their features. After passing FPS and k-NN, the trigonometric functions  $PosE(\cdot)$  are used to reveal the local features simply. Specifically, for each centroid  $p_c$  and its neighbourhood  $p_i$ , Local Geometry Aggregation (LGA) is applied to implement feature extraction. The specific process of LGA is as follows: first,  $p_c$  and  $p_i$  are concatenated along the feature dimension to assign a large receptive field to each point feature and expand the feature. Second,  $PosE(\cdot)$ , which refers to position encoding in the Transformer, can effectively encode the relative position information. The expanding feature combines  $PosE(\cdot)$  to contain the local geometry information. Finally, pooling operations are used to aggregate the expanding feature. After the multi-stage hierarchy, both max and average pooling are performed to aggregate the local structure feature  $f_{pb}$ .

Convolution-based processing. Drawing on the idea of point cloud gridding [8] that has developped in recent years, we grid the input point cloud to extract its global contour features. The point cloud is regularized using a 3D grid as an intermediate representation, whereby an unordered and irregular point cloud is converted into a regular 3D grid denoted as  $G = \langle V, W \rangle$ . This conversion ensures the preservation of the spatial layouts of the point cloud, with each point  $p_i \in \mathbb{R}^3$  being assigned to the vertex set *V*, and corresponding values are stored in the set *W*. As illustrated in Figure 2, a cell is defined as a cube composed



Figure 2. Gridding processing.

Next, the objective of the 3D Convolutional Neural Network (3D CNN) with skip connections is to extract the global contour information from a 3D grid. The architecture of the 3D CNN includes four 3D convolutional layers, each composed of a batch normalization layer, an activation function, and a max pooling layer. Finally, a shared MLP is used to output the global contour feature  $f_{cb}$  representing the feature obtained from convolution-based processing.

#### 2.2. Decoder

By extracting point cloud features from the encoder, we obtained  $f_{pb}$  and  $f_{cb}$ , whose sizes are  $a \times C$  and  $b \times C$ , respectively, where a and b are weight coefficients. In the attention module, we first concatenated the two features along the feature dimensions, and extended the concatenated feature dimensions, denoted as  $f_{pb-cb}^{expand}$ , in order to expand the receptive field to increase the representational capability. Second, the extended feature is input into the max-pooling MLP pipeline (*MaxpoolMLP*) and the average-pooling MLP pipeline (*AvgpoolMLP*), respectively, to obtain the weighted point cloud features. Then, based on the weight values, the  $1 \times C$  features with the highest weights are used to represent the global features  $f_G$  of the input point cloud. The experimental results show that although the structure of the attention module is simple, the effect is significantly improved.

$$f_G = Topk \left\{ MaxpoolMLP \left( f_{pb-cb}^{expand} \right) + AvgpoolMLP \left( f_{pb-cb}^{expand} \right) \right\}$$
(1)

Next, we generate the complete and dense point cloud from the global feature  $f_G$ . In the first step, a coarse point cloud is generated by passing  $f_G$  through an MLP and transforming the output into a  $C \times 3$  matrix. In the second step, for each point  $q_i$  in the coarse point cloud, a patch of  $t = u^2$  points in local coordinates centered at  $q_i$  is generated using the folding operation. Subsequently, these points are transformed into global coordinates by adding  $q_i$  to the output, where u represents the side lengths of the 2D grid. Combining all C patches generates a complete point cloud consisting of  $n = C \times t$  points. This two-step process enables FuNet to generate a complete point cloud using fewer parameters compared to a fully connected decoder, while also offering greater flexibility than a folding-based decoder.

# 2.3. Loss Function

The loss function is used to evaluate the disparity between the ground truth point cloud and the output point cloud. Given the unordered nature of point clouds, the loss function must be permutation-invariant. Common choices for point cloud completion loss functions include Chamfer Distance (CD) [20] and Earth Mover's Distance (EMD) [20]. Due

to the high memory requirements of EMD, with a complexity of  $O(n^2)$ , and considering that the number of the reconstructed points must be equal to the number of points in the ground truth point cloud, CD with a complexity of O(nlogn) is chosen in our experiment. In addition, Uniform Loss [23] is incorporated to enhance the uniformity of the output point cloud.

Chamfer Distance: By definition, Chamfer Distance denotes the sum of the average closest distance from a point in the output point cloud  $S_1$  to a point in the ground truth point cloud  $S_2$ , and the average closest distance from a point in  $S_2$  to a point in  $S_1$ .

$$CD(S_1 - S_2) = \mathcal{L}_{S_1 - S_2} + \mathcal{L}_{S_2 - S_1} = \frac{1}{N_{S1}} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{N_{S2}} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2$$
(2)

where  $\mathcal{L}_{S_1-S_2}$  denotes the average distance from the point of  $S_1$  to the closest point of  $S_2$ , and  $\mathcal{L}_{S_2-S_1}$  denotes the average distance from the point of  $S_2$  to the closest point of  $S_1$ .  $N_{S1}$  and  $N_{S2}$  are the numbers of points for  $S_1$  and  $S_2$ , respectively.

In general, the loss function CD has two forms, CD - l1 and CD - l2, which are defined as follows:

$$\mathcal{L}_{CD-l1}(S_1 - S_2) = \left(\sqrt{\mathcal{L}_{S_1 - S_2}} + \sqrt{\mathcal{L}_{S_2 - S_1}}\right)/2 \tag{3}$$

$$\mathcal{L}_{CD-l2}(S_1 - S_2) = \mathcal{L}_{S_1 - S_2} + \mathcal{L}_{S_2 - S_1}$$
(4)

They are both used in the loss function of the network.

Uniformity Loss: Uniformity is usually used to evaluate the homogeneity of the complete point cloud distribution, and it is expressed as:

$$\mathcal{L}_{uni} = \sum_{j=1}^{M} U_{imbalance} \left( S_j \right) \cdot U_{clutter} \left( S_j \right)$$
(5)

where  $S_j$  is the subset of points ( $j = 1, \dots, M$ ) obtained by cropping from the output point cloud using farthest point sampling and a ball query with radius  $r_d$ . Here,  $U_{clutter}$  considers local distribution uniformity, while  $U_{imbalance}$  considers non-local uniformity to encourage better point coverage.

$$U_{imbalance}\left(S_{j}\right) = \frac{\left(\left|S_{j}\right| - \hat{n}\right)^{2}}{\hat{n}}$$

$$\tag{6}$$

where  $\hat{n}$  is the expected number of points in  $S_j$  and the chi-square test is employed to quantify the bias of  $|S_j|$  from  $\hat{n}$ .

$$U_{clutter}(S_{j}) = \sum_{k=1}^{|S_{j}|} \frac{\left(d_{j,k} - \hat{d}\right)^{2}}{\hat{d}}$$
(7)

where  $d_{j,k}$  represents the distance to the nearest neighbor for the *k*-th point in  $S_j$ , and  $\hat{d}$  is approximately calculated as  $\sqrt{\frac{2\pi r_d^2}{|S_j|\sqrt{3}}}$  (assuming  $S_j$  has a uniform distribution). The chi-square test is employed once again to quantify the bias of  $d_{j,k}$  from  $\hat{d}$ .

The loss function  $\mathcal{L}$  that we propose is as follows, where  $\alpha$ ,  $\beta$  and  $\gamma$  are weight coefficients.

$$\mathcal{L} = \alpha \mathcal{L}_{CD-l1} \left( P_{pb} - P_{gt} \right) + \alpha \mathcal{L}_{CD-l1} \left( P_{cb} - P_{gt} \right) + \beta \mathcal{L}_{CD-l1} \left( P_{complete} - P_{gt} \right) + \gamma \mathcal{L}_{uni} \left( P_{complete} \right)$$
(8)

Of these, the first term of the function evaluates the CD - l1 loss between the coarse point cloud  $P_{pb}$  generated by  $f_{pb}$  and the ground truth point cloud. Similarly, the second term of the function evaluates the loss between the coarse point cloud  $P_{cb}$  generated by  $f_{cb}$  and the ground truth point cloud, and the third term evaluates the loss between the

complete point cloud  $P_{complete}$  and the ground truth point cloud. The last term evaluates the uniformity of the complete point cloud  $P_{complete}$ .

## 3. Experiments

# 3.1. Dataset

The ShapeNet dataset used for point cloud completion is derived from PCN [13], comprising 30,974 3D models distributed across eight categories. Each model's ground truth point cloud, containing 16,384 points, is uniformly sampled on mesh surfaces. The partial point cloud is generated by back-projecting a 2.5D depth map into 3D, simulating data captured by real sensors, with each partial point cloud containing no more than 1024 points. The distribution of the ShapeNet dataset is shown in Table 2.

Categories	<b>Training Set</b>	Validation Set	Test Set	
Airplane	3795	100	150	
Ċar	5677	100	150	
Table	5750	100	150	
Chair	5750	100	150	
Lamp	2068	100	150	
Cabinet	1322	100	150	
Sofa	2923	100	150	
Vessel	1689	100	150	

Table 2. The numbers in the training set, validation set and test set for the ShapeNet dataset.

#### 3.2. Evaluation Metrics

The evaluation metrics in the experiment are CD [20] and F-score [21].

The F-score evaluates the percentage of correctly reconstructed points, which is defined as the harmonic mean between precision and recall. Precision quantifies the ratio of reconstructed points within a constant distance to the ground truth, reflecting the accuracy of the reconstruction. Similarly, recall quantifies the ratio of points on the ground truth within a constant distance to the reconstruction, reflecting the completeness of the reconstruction. The distance threshold d can be adjusted to control the strictness of the F-score. The F-score is defined as follows:

$$F-score(d) = \frac{2P(d)R(d)}{P(d)+R(d)}$$
(9)

where P(d) and R(d) represent the precision and recall for a distance threshold *d*, respectively. The mathematical expressions of P(d) and R(d) are defined as follows:

$$P(d) = \frac{1}{N_{S1}} \sum_{x \in S_1} [\min_{y \in S_2} ||x - y|| < d]$$
  

$$R(d) = \frac{1}{N_{S2}} \sum_{y \in S_2} [\min_{x \in S_1} ||y - x|| < d]$$
(10)

where  $S_1$  is the reconstructed point cloud set and  $S_2$  is the ground truth point cloud set.  $N_{S1}$  and  $N_{S2}$  are the numbers of points for  $S_1$  and  $S_2$ , respectively.

CD can be used to evaluate the similarity between the ground truth and the output, and the F-score can be used to evaluate the precision and recall between them. Combining them can help to effectively evaluate the results of point cloud completion.

#### 3.3. Implementation Details

In our experiment, all models are trained for 200 epochs, with a batch size of 24, a learning rate of  $1 \times 10^{-4}$  (decaying by 0.5 every 40 epochs), and an Adam optimizer. The networks are trained using the PyTorch framework on NVIDIA RTX A4000 GPU and running on Ubuntu 20.04. Figure 3 shows that FuNet is trained in 200 epochs on the

ShapeNet dataset with CD - l1 loss evaluation metrics and converges within 200 epochs. The number of input and output points in the point cloud shall not exceed 1024 and 16,384, respectively. And it can be found that the CD - l1 loss reached a constant after 120 epochs and without overfitting at the end of the epochs.



**Figure 3.** FuNet is trained in 200 epochs on the ShapeNet dataset with the CD - l1 loss evaluation metric and converges within 200 epochs.

More details about FuNet's parameters and hyperparameters are given below: in the point-based processing, multi-stage hierarchy has four stages, and the *k* of *k*-NN is 8. In the convolution-based processing, the size of the grid resolution is  $64^3$  and the architecture of the 3D CNN includes four 3D convolutional layers. The radio of  $f_{pb}$  and  $f_{cb}$  is a : b = 1 : 2. We set  $\alpha$  as [1.0, 0.7, 0.5, 0.5],  $\beta$  as [0.01, 0.1, 0.5, 1.0], and  $\gamma$  as [0.01, 0.1, 0.1, 0.1] in the loss function  $\mathcal{L}$ , where the weights are changed at 10,000 train steps, 20,000 train steps, and 50,000 train steps, respectively.

#### 3.4. Completion Results on the ShapeNet Dataset

To verify the effectiveness of FuNet, it was tested on the ShapeNet dataset and compared to relative completion networks. PCN [13] uses PointNet to extract the global feature and outputs a complete point cloud through the fully-connected operation and foldingbased operation. FoldingNet [10] serves as a basic method utilized in PCN, where it deforms a 128 × 128 2D grid into a 3D point cloud. GRNet [8] introduces 3D grids as intermediate representations to regularize unordered point cloud.

Figure 4 shows a comparison of the visualization results of different completion networks, from which the following advantages of FuNet can be summarized. (1) For the complete point cloud generated by FuNet, it has a higher and more homogeneous point cloud density, and smoother and more complete global contours. Also, on local structures, such as the legs of a table or chair, it is more similar to the ground truth. (2) It has good generalization ability in the face of different categories of point cloud data.



Figure 4. Comparison of the visualization results of different networks on the ShapeNet testing set.

In Tables 3 and 4, the experimental result shows that FuNet provides the best performance in terms of CD and F-score in most categories. As shown in Table 3, FuNet is superior to all compared networks for all categories of CD, except for the lamp category, where CD is slightly higher than that of GRNet. Its average CD is 5.17% lower than the optimal network, GRNet. As shown in Table 4, FuNet is superior to all compared networks for all categories of F-score, except for the lamp category and the chair category. Its average F-score is 4.75% higher than that of the optimal network, PCN.

Table 3. Poir	nt complet	ion results or	ι the ShapeNo	et dataset ı	using CD -	- <i>l</i> 1 compute	d on 16,384 p	oints
and multipli	led by $10^3$ .	The best res	ults are highl	ighted in b	old (lower	r is better).		

Categories	FoldingNet	PCN	GRNet	FuNet
Airplane	9.69	6.35	7.18	5.78
Čar	12.16	9.13	10.36	8.96
Table	13.54	9.84	9.67	9.54
Chair	16.55	12.03	11.86	10.46
Lamp	15.99	14.52	9.69	12.97
Cabinet	16.59	12.82	11.82	11.07
Sofa	16.81	14.46	13.78	11.31
Vessel	12.33	10.16	9.24	9.17
Average	14.21	11.16	10.45	9.91

It can be seen that the CD and F-score of FuNet in the lamp category are normally not optimal. This could be attributed to two possible reasons. On the one hand, certain objects within this category, such as brackets or rods, may contain very thin structures, and it is difficult to deform a 2D grid into thin structures. On the other hand, the lamp category includes a wide variety of types and shapes, leading to significant variations in their geometries.

Overall, the comprehensive evaluation shows FuNet's remarkable effectiveness in point cloud completion tasks, with superior performance in various categories compared

to relative networks. FuNet's ability to generate dense, smooth and uniform point clouds demonstrates its potential for real-world applications.

Categories	FoldingNet	PCN	GRNet	FuNet
Airplane	0.623	0.863	0.828	0.871
Čar	0.439	0.617	0.608	0.695
Table	0.390	0.608	0.621	0.675
Chair	0.222	0.583	0.540	0.578
Lamp	0.255	0.579	0.684	0.532
Cabinet	0.205	0.534	0.559	0.564
Sofa	0.202	0.596	0.439	0.698
Vessel	0.459	0.666	0.662	0.677
Average	0.349	0.631	0.617	0.661

**Table 4.** Point completion results on the ShapeNet dataset using F-score (0.01) computed on 16,384 points. The best results are highlighted in bold (higher is better).

## 3.5. Completion Result on Small Sample Dataset

In addition, the effectiveness was verified in the case of small sample dataset using the ShapeNet dataset by category. The testing dataset had 250 samples for each category. The experiment results in Table 5 show that FuNet achieved better completion in most of the small sample datasets. It shows that FuNet can maintain good performance even with small samples in specific application scenes. The multi-feature fusion module of FuNet ensures excellent results with a small number of samples. Compared to PCN, FuNet achieves better results in most categories. Figure 5 shows the model comparison results and their details; the results of FuNet show better surface smoothing and a more uniform geometric structure.

**Table 5.** Training and testing by category on ShapeNet dataset using CD - l2 computed on 16,384 points and multiplied by  $10^4$ . The best results are highlighted in bold (lower is better).

Categories	PCN	FuNet
Airplane	2.154	1.635
Ĉar	2.998	2.692
Table	5.844	6.949
Chair	5.979	5.808
Lamp	8.964	9.522
Cabinet	5.504	4.620
Sofa	6.675	6.487
Vessel	4.486	3.888



Figure 5. Model comparison results and their details on a small sample dataset.

#### 3.6. Ablation Study

To further verify the effectiveness and applicability of point-based processing, convolutionbased processing, and the attention module, we conducted an ablation experiment on FuNet using the whole ShapeNet dataset.

For the point-based processing and the convolution-based processing modules, only one of them was utilized to extract point cloud features. In the absence of the attention module, simple feature concatenation was employed instead. The different models for the ablation experiments are specified below: [A] utilized only a convolution-based processing module without an attention module. [B] utilized only a point-based processing module without an attention module. It is noted that the global features of the point cloud may not be adequately accommodated using only one feature extraction method. [C] utilized both the point cloud feature extraction methods and directly concatenated the features, then used an MLP to extract the global feature. [D] represents the complete FuNet model.

For the four models, their parameters and hyperparameters are similar to the complete FuNet when used in point-base processing and convolution-based processing. [A] output  $f_{cb}$  directly in the encoder and [B] output  $f_{pb}$  directly in the encoder as a global feature, whose size was  $1 \times C$ . [C] output the global feature with the same size.

In Table 6, the attention module decreased the CD by 15.95% and increased the F-score by 8.01%, which means that the attention module can extract the most important features. The comparison reveals that only when all the proposed modules are present will the completion capability of the network be optimal.

best results highlighted in bold. Point-based, convolution-based and attention in conditions indicate the presence or absence of the relevant module in the models, and are ticked if present (CD—lower is better, F-score—higher is better).

**Table 6.** Comparison of the results of ablation experiments using CD - l1 and F-score (0.01), with the

M. 1.1.	Conditions		CD	T.C.	
Models	Point-Based	Convolution-Based	Attention	CD	F-Score
[A]				12.34	0.593
[B]				12.54	0.564
[C]		$\checkmark$		11.79	0.612
[D]			$\checkmark$	9.91	0.661

## 4. Conclusions

In addressing the challenge of sparse and incomplete sampling in real-world objects and to satisfy the requirements of downstream tasks, we introduce FuNet, a novel point cloud completion network designed to transform a partial point cloud into a complete point cloud. By employing both point-based and convolution-based processing, our approach captures local structural features and global contour features crucial for accurate completion. In addition, the integration of an attention module facilitates effective feature fusion through weighted aggregation. Finally, a coarse-to-fine decoder converts the coarse point cloud into a complete and dense point cloud.

Our comprehensive evaluation undertaken through an ablation study reveals that the integration of these modules leads to a significant enhancement in point cloud completion performance. Comparing the optimal completion networks, FuNet decreases the CD by 5.17% and increases the F-score by 4.75% on the whole ShapeNet dataset. Moreover, across various object categories and particularly on small sample datasets, FuNet mostly outperforms other methods, demonstrating its robustness and applicability. These results confirm the effectiveness and versatility of our approach, which holds promise for diverse applications across different sensors and object types.

With the current outstanding performance of FuNet, we aim to explore further enhancements in terms of speed and robustness. Additionally, we plan to integrate downstream tasks such as segmentation and classification models to broaden FuNet's functionality, making it more adaptable to various application scenarios. At the same time, FuNet can be used in pre-processing for 3D object detection, such as in autonomous driving, to improve object detection performance.

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

**Funding:** Guangdong Basic and Applied Basic Research Foundation (2023A1515011590), Science and Technology Projects in Guangzhou (202201010540).

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

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

# References

- 1. Dandois, J.P.; Olano, M.; Ellis, E.C. Optimal Altitude, Overlap, and Weather Conditions for Computer Vision UAV Estimates of Forest Structure. *Remote Sens.* 2015, *7*, 13895–13920. [CrossRef]
- 2. Pérez, L.; Rodríguez, Í.; Rodríguez, N.; Usamentiaga, R.; García, D.F. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors* **2016**, *16*, 335. [CrossRef] [PubMed]
- Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings
  of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
- 4. Qi, C.R.; 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 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018), Montreal, QC, Canada, 3–8 December 2018.
- 6. 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]
- Xu, M.; Ding, R.; Zhao, H.; Qi, X. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3173–3182.
- Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; Sun, W. Grnet: Gridding residual network for dense point cloud completion. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 365–381.
- Wang, X.; Ang, M.H.; Lee, G.H. Voxel-based network for shape completion by leveraging edge generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13189–13198.
- Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 206–215.
- 11. Liu, M.; Sheng, L.; Yang, S.; Shao, J.; Hu, S.-M. Morphing and sampling network for dense point cloud completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11596–11603.
- Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. Pointr: Diverse point cloud completion with geometry-aware transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 12498–12507.
- 13. 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.
- 14. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning representations and generative models for 3d point clouds. In Proceedings of the International Conference on Machine Learning, Macau, China, 26–28 February 2018; pp. 40–49.
- Wang, Y.; Tan, D.J.; Navab, N.; Tombari, F. Softpoolnet: Shape descriptor for point cloud completion and classification. In *Computer Vision–ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part III 16; Springer Nature: Berlin/Heidelberg, Germany, 2020; pp. 70–85.
- 16. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. Pf-net: Point fractal network for 3d point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
- Zhang, J.; Chen, X.; Cai, Z.; Pan, L.; Zhao, H.; Yi, S.; Yeo, C.K.; Dai, B.; Loy, C.C. Unsupervised 3d shape completion through gan inversion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1768–1777.

- Xiang, P.; Wen, X.; Liu, Y.-S.; Cao, Y.-P.; Wan, P.; Zheng, W.; Han, Z. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 5499–5509.
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
- Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
- Tatarchenko, M.; Richter, S.R.; Ranftl, R.; Li, Z.; Koltun, V.; Brox, T. What do single-view 3d reconstruction networks learn? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3405–3414.
- Zhang, R.; Wang, L.; Wang, Y.; Gao, P.; Li, H.; Shi, J. Parameter is Not All You Need: Starting from Non-Parametric Networks for 3D Point Cloud Analysis. arXiv 2023, arXiv:2303.08134.
- 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 IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7203–7212.

**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.