applied sciences

MDPI

Article

An Efficient Bidirectional Point Pyramid Attention Network for 3D Point Cloud Completion

Yang Li *, Yao Xiao, Jialin Gang and Qingjun Yu

School of Digital Arts and Design, Dalian Neusoft University of Information, Dalian 116023, China
* Correspondence: liyang_sy@neusoft.edu.cn

Abstract: Point cloud completion is a necessary task in real-world applications of recovering a complete geometry from missing regions of 3D objects. Furthermore, model efficiency is of vital importance in computer vision. In this paper, we present an efficient encoder–decoder network that predicts missing point clouds on the basis of incomplete point clouds. There are several advantages to this approach. First, a Mixed Attention Module (MAM) was implemented to obtain the correlational information of points. Second, the proposed Bidirectional Point Pyramid Attention Network (BiPPAN) can achieve simple and fast multiscale feature fusion to capture important features. Lastly, the designed encoder–decoder framework comprises skip connections to capture long-distance dependencies and structural information. We can conclude from the results of the experiments that the proposed network is an efficient and effective method to accomplish point cloud completion tasks.

Keywords: 3D point cloud; shape completion; mixed attention mechanism; bidirectional point pyramid attention network

check for updates

Citation: Li, Y.; Xiao, Y.; Gang, J.; Yu, Q. An Efficient Bidirectional Point Pyramid Attention Network for 3D Point Cloud Completion. *Appl. Sci.* **2023**, *13*, 4897. https://doi.org/10.3390/app13084897

Academic Editors: João M. F. Rodrigues and Rubén Usamentiaga

Received: 18 November 2022
Revised: 17 December 2022
Accepted: 11 January 2023
Published: 13 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Three-dimensional vision is a major research area in computer vision. Recently, due to the rapid development of 3D sensing technology, 3D vision has been one of the research hotspots. Three-dimensional data can emerge in different forms, including depth images, point clouds, meshes, and volumetric grids. The point cloud is a common display form because it preserves the original geometric information in 3D space and requires less memory to store. However, point clouds obtained from laser scanners or other devices are usually incomplete, which brings some difficulties to the subsequent processing of point clouds. Therefore, as a point cloud preprocessing method, completing point clouds from missing and sparse raw data has become an important task [1,2].

Deep learning technology has dominated numerous research areas and shown significant advantages in image recognition [3], speech recognition [4], natural language processing [5] and other fields [6]. With the widespread application of 3D data, many available large 3D datasets have emerged, such as ModelNet [7] and ShapeNet [8]. At the same time, deep learning on 3D shape classification, 3D object detection and tracking, and 3D point cloud completion is receiving increasing attention [9,10].

As a pioneering work, PointNet [11] developed by Qi et al. directly takes point clouds as input and output that respect the permutation invariance of points. The major contribution is that PointNet uses a max pooling layer as a symmetric function to extract global features from all the points, and multi-layer perceptron (MLP) layers to learn pointwise features independently. It is able to carry out object classification, part segmentation, and scene semantic parsing. After PointNet, many other methods sprang up to generate complete point clouds [12,13]. The challenge of point cloud completion is to extract structural information from unordered and unstructured point cloud data. Obviously, learned features directly identify the quality of completion. Therefore, it is very important to effectively extract and exploit the features of point clouds to raise the accuracy of prediction. We set the

goal of capturing the correlational information of points and selectively convey geometric information from the local regions of incomplete point clouds, so that we could reconstruct complete point clouds. The designed modules enable our model to learn structural features better, and preserve detailed information for point cloud completion to improve prediction accuracy.

In this work, we demonstrate a method to only generate the missing part of point clouds. Our main contributions are reflected in three aspects and are summed up in Figure 1.

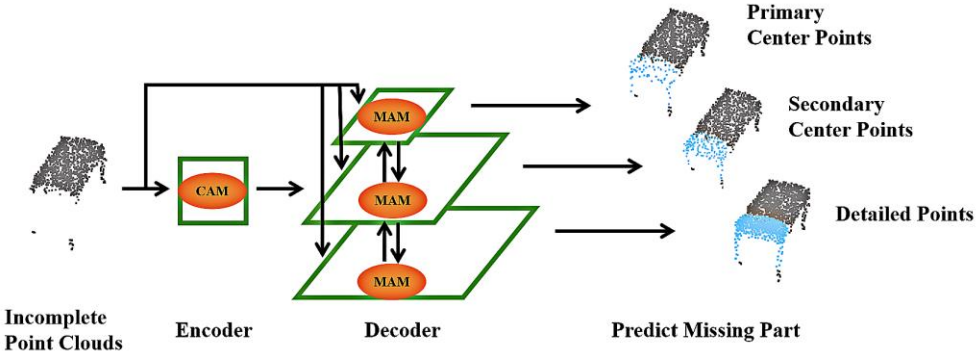


Figure 1. Our method directly predicts missing point clouds (blue points) with incomplete point clouds (gray points) as input. We designed an encoder–decoder architecture with a Channelwise Attention Module (CAM) and a Mixed Attention Module (MAM). The decoder utilizes a bidirectional point pyramid attention network and skip connections to generate multiscale predictions (primary center points, secondary center points, detailed points).

(1) We offer an encoder–decoder network for point cloud completion. A set of experiments proved the effectiveness and robustness of the proposed method on several challenging datasets.
(2) We applied a Channelwise Attention Module (CAM) and a Mixed Attention Module (MAM) to introduce a reasonable weight to learn the importance of distinct features. While fusing different input features, the contributions of the fused output features are not equal. The attention mechanism allows for the network to infer the missing regions from incomplete point clouds, exploiting more effective geometric information.
(3) We designed a simple and highly effective decoder that consists of skip connections and a Bidirectional Point Pyramid Attention Network (BiPPAN). BiPPAN applies top–down and bottom–up multiscale feature fusion. In fact, a multilayer pyramid network model was used on PF-Net [14]. However, a point cloud generation method propagating low-level features to improve the entire feature hierarchy has not yet been invented.

To verify whether the proposed network acted effectively, we evaluated our network with extensive experiments on public ShapeNet datasets of three different sizes (ShapeNet-13, ShapeNet-16, ShapeNet-55). We can conclude from the experimental results that our proposed method outperformed PF-Net as the baseline method, whether it was on large or small datasets.

The remainder of the paper is organized as follows. Section 2 demonstrates the related works. Section 3 presents our network and loss function. Section 4 describes our experiments on the ShapeNet dataset to perform completion in point clouds. Lastly, Section 5 summarizes our paper.

2. Related Work

2.1. 3D Shape Completion

In this section, we review recent work applying deep learning technology to shape completion, and then introduce the feature pyramid network and its application to point cloud completion.

In the field of shape completion, PCN [15] is the first deep learning network using point clouds without any voxelization. It presents an encoder–decoder architecture and FoldingNet to generate a dense point cloud in a coarse-to-fine fashion. Subsequently, many models have been developed to achieve high-resolution completion and strong generalization performance. By combining reinforcement learning with a generative adversarial network (GAN), RL-GAN-Net [16] and Render4Completion [17] are a novel set of architectures for 3D point cloud representation learning and generation whose reconstruction results had better robustness. Huang et al. [18] proposed a new recurrent forward network (RFNet) for point cloud completion and achieved the most advanced performance, which consisted of three separate modules: recurrent feature extraction (RFE), forward dense completion (FDC), and raw shape protection (RSP).

2.2. Multiscale Feature Representations

The feature pyramid network (FPN) is a top–down pathway to combine multiscale features. It can effectively represent multiscale features and is widely used in various tasks [19,20]. Huang et al. [14] designed a point fractal network (PF-Net) to hierarchically generate a missing point cloud that utilizes a multilayer point pyramid decoder (PPD) to predict the completion results of different layers.

3. Methods

In this part, we state the detailed information of our network. Figure 2 shows the full architecture of our method, which only predicts the missing parts from incomplete inputs. It consists of an encoder with Attention Combined Multi-Layer Perceptron (ACMLP) and a decoder with a Bidirectional Point Pyramid Attention Network (BiPPAN). The attention module was applied to the encoder and decoder as detailed in Section 3.1. The encoder shows its talent not only with excellent feature extraction, but also in the presentation of feature aggregation. We detail the point feature encoder in Section 3.2. Between the input and the decoder, skip connections connect the geometric information of the point cloud with the point features in the decoder. BiPPAN performs the prediction of missing point clouds with different resolutions. We describe the decoder in detail in Section 3.3. To optimize the network parameters, multistage completion loss was applied. The multistage completion loss was set as the Chamfer Distance (CD) error between the predicted point cloud and ground truth, which was composed of three items. We detail the loss function in Section 3.4.

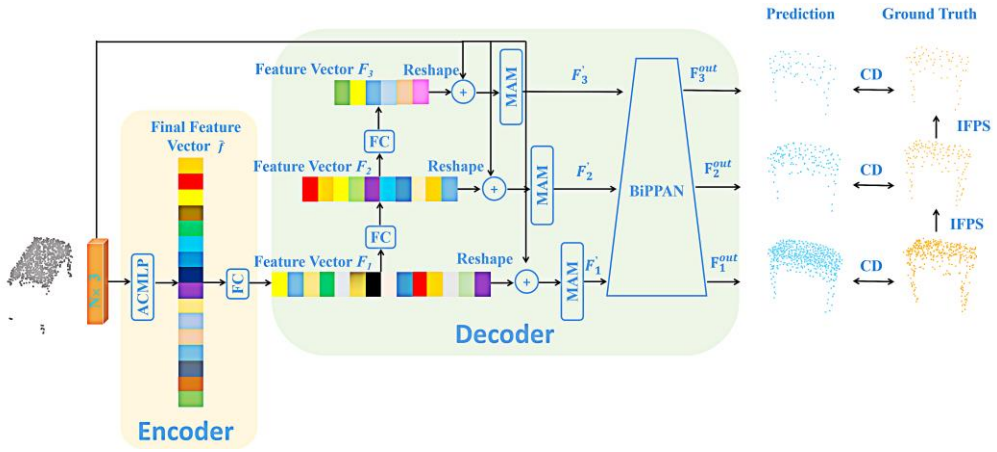


Figure 2. Illustration of our framework. The encoder (yellow) adopts ACMLP to extract features from input point clouds. The decoder (green) adopts BiPPAN with skip connections to predict the multiscale missing parts. Then, the Chamfer Distance errors of the prediction point cloud and ground truth are calculated.

3.1. Mixed Attention Module

Because of its ability to select features, the attention mechanism has been applied in a series of tasks, such as machine translation, object recognition, visual question answering and so on [21,22]. Recent studies showed that the attention mechanism has been widely used in 3D shape completion. Wen et al. [23] proposed a skip-attention network (SA-NET) for point cloud completion tasks, and designed a skipping-attention mechanism to effectively utilize the local structural details of a point cloud. Yu et al. [24] presented a novel model called PoinTr for point cloud completion that fully uses an encoder–decoder architecture with the adapted transformer blocks. Inspired by squeeze-and-excitation networks (SENets) [25] that adaptively recalibrate channelwise feature responses, we adopted channelwise and pointwise attention in our encoder–decoder architecture to conquer the problem of point cloud completion. The structure of the mixed attention module is illustrated in Figure 3.

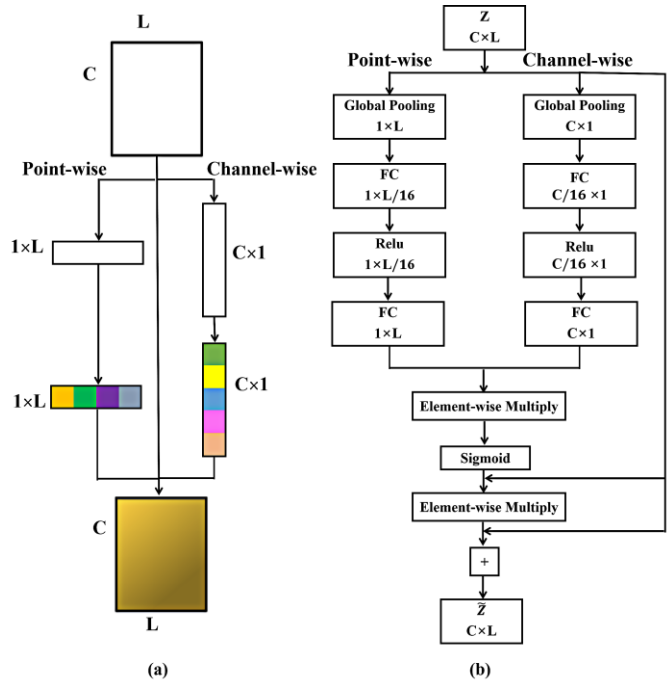


Figure 3. (**a**) Diagram and (**b**) schema of the mixed attention module.

Channelwise attention: Given a point feature matrix z with the size of $C \times L$, we applied channelwise average pooling to obtain a global feature vector with the size of $C \times 1$. Then, we operated two fully connected (FC) layers around ReLU activation units. The layers were able to fully capture channelwise dependencies.

$$CA = F(z, W) = W_2\delta(W_1 pool(z)) \quad (1)$$

where the pool denotes the global average pooling, $W_1 \in \mathbb{R}^{C/16 \times C}$ and $W_2 \in \mathbb{R}^{C \times C/16}$ denote parameters in the two FC layers, and δ refers to the ReLU activation units. $CA \in \mathbb{R}^{C \times 1}$ is the channelwise attention of z.

Pointwise attention: Similar to the strategy for predicting the channelwise attention for a point feature matrix $z \in \mathbb{R}^{C \times L}$, a pointwise average pooling operation was performed to aggregate the pointwise feature vector with the size of $1 \times L$. Then, we computed $PA = F'(z, W') = W_2'\delta(W_1' pool(z))$. The weight parameters of two fully connected layers are $W_1' \in \mathbb{R}^{L/16 \times L}$ and $W_2' \in \mathbb{R}^{L \times L/16}$. $PA \in \mathbb{R}^{1 \times L}$ is the pointwise attention of z.

Lastly, the output feature vector $\widetilde{z}$ of the mixed attention module was obtained by weighting the feature z with CA and PA as follows. Using the sigmoid function σ, the attention matrix was normalized to [0,1].

$$\widetilde{z} = \sigma(CA \times PA) \times z + z \quad (2)$$

Through the operations above, the feature representation $\widetilde{z}$ could learn discrimination and robust representation for point clouds.

3.2. Point Feature Encoder

The size of the incomplete point cloud input was N = 2048 with its 3-dimensional coordinates. The encoder network was utilized to extract features from the incomplete input. Motivated by PointNet, we designed an Attention Combined Multi-Layer Perceptron (ACMLP) as our point cloud feature encoder. As shown in Figure 4, the structure of the encoder comprised a shared multi-layer perceptron (MLP) with LeakyReLU activation, which consisted of five layers with neuron sizes 64, 128, 256, 512, and 1024. Different dimensions of MLP can extract low, mid-, and high-level features, each of which contains rich point cloud information. In order to utilize these features effectively, we used max pooling to obtain the global latent representation f_i of the output of the last four layers, whose size was $f_i := 128, 256, 512, 1024$, for i = 1,...,4, and concatenate f_i to produce a 1920-dimensional feature vector. Then, we adopted a channelwise attention module for the 1920-dimensional features to obtain the weighted multilevel features $\widetilde{f}$ that contained low-, mid-, and high-level feature information, and could help in generating effective features for point cloud prediction. Since we used max pooling for the 1920-dimensional feature vector, we did not take the pooling layer into the channelwise attention module. Eventually, we generated the final feature vector F_1 through a fully connected layer.

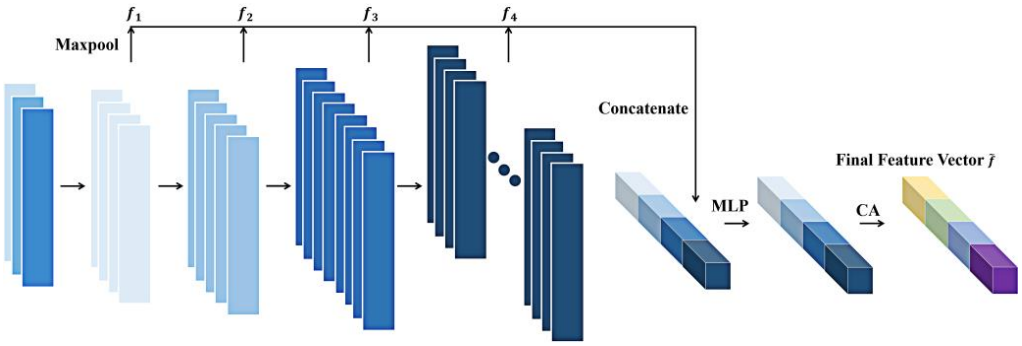


Figure 4. Structure of the proposed ACMLP.

3.3. Point Pyramid Decoder

The decoder structure is composed of two modules: skip connections and Bidirectional Point Pyramid Attention Network (BiPPAN).

As shown in Figure 2, we concatenated the input incomplete point clouds with the $F_1 - F_3$ via the long-range skip connections. We obtained F_2 and F_3 by passing F_1 through fully connected layers. We also employed the mixed attention module to fuse these features and generate $F'_1 - F'_3$ as the input of BiPPAN. There are two benefits to such skip connections. One is to provide long-range information compensation, so the raw incomplete point cloud geometry information is still available in the decoder architecture. The other is that residual learning can facilitate gradient backpropagation.

BiPPAN is based on the feature pyramid network (FPN) that contains a top–down pathway to aggregate multiscale features from Level 1 to 3 ($F'_1 = 512 \times 3$, $F'_2 = 256 \times 3$ and $F'_3 = 128 \times 3$), as shown in Figure 5a. Levels 1–3 represent low-to-high levels. The conventional FPN fuses multiscale features in a top–down manner. Formally,

$$F_3^{out} = \mathrm{Conv}(F'_3) \tag{3}$$

$$F_2^{out} = \mathrm{Conv}(F'_2 + \mathrm{Resize}(F_3^{out})) \tag{4}$$

$$F_1^{out} = \mathrm{Conv}(F'_1 + \mathrm{Resize}(F_2^{out})) \tag{5}$$

where Resize is an upsampling/downsampling or reshape operation for size matching, and Conv is a convolutional operation. F_1^{out}, F_2^{out} and F_3^{out} ($F_1^{out} = 512 \times 3$, $F_2^{out} = 128 \times 3$ and $F_3^{out} = 64 \times 3$) are the predicted missing regions of the incomplete point cloud in different resolutions.

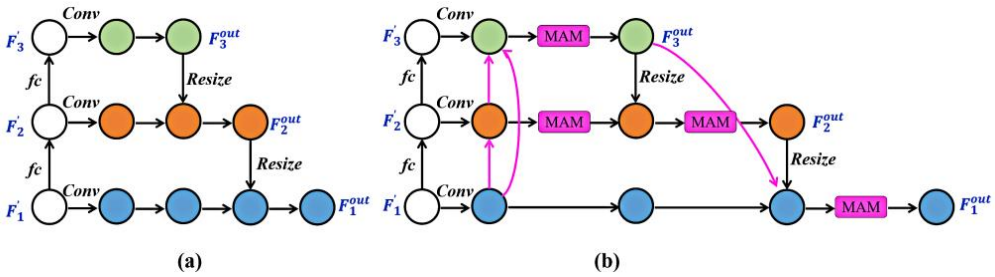


Figure 5. Comparison between (**a**) FPN and (**b**) the proposed BiPPAN decoder.

Following this idea, we designed a bidirectional point pyramid attention network to optimize multiscale feature fusion on top of FPN, as depicted in Figure 5b. First, we added not only a bottom–up path aggregation network, but also cross-level connections. By fully fusing high-level features with rich global information and low-level features with many fine details of local geometry information, we could predict the detailed structure of the missing point cloud. Second, we introduced the mixed attention module to learn the importance of different input features whose purpose is to selectively aggregate the features.

3.4. Loss Function

The loss function in our method is multistage completion loss L_{com}. Because we set the pyramid level N = 3 in our network, the multistage completion loss L_{com} was composed of three parts (d_{CD1}, d_{CD2}, d_{CD3}):

$$L_{com} = d_{CD1}(F_1^{out}, F_{gt}) + \alpha \times d_{CD2}\left(F_2^{out}, F'_{gt}\right) + 2\alpha \times d_{CD3}\left(F_3^{out}, F''_{gt}\right) \tag{6}$$

where α is the hyperparameter, and d_{CD} is the Chamfer Distance proposed by Fan [26]. The mean Chamfer Distance can measure the average nearest squared distance between prediction point cloud F^{out} and ground truth F_{gt}, which is calculated with:

$$d_{CD}(F^{out}, F_{gt}) = \frac{1}{F^{out}} \sum_{x \in F^{out}} \min_{y \in F_{gt}} \|x - y\|_2^2 + \frac{1}{F_{gt}} \sum_{y \in F_{gt}} \min_{x \in F^{out}} \|y - x\|_2^2 \tag{7}$$

In detail, d_{CD1} calculates the squared distance between predicted detailed points F_1^{out} and the ground truth of the missing region F_{gt}. Items d_{CD2} and d_{CD3} calculate the squared distance among secondary central points F_2^{out}, predicted primary central points, F_3^{out} and ground truth F'_{gt}, F''_{gt}.

4. Experimental

In this part, we present the three benchmarks for point cloud completion and the implementation details of our network. Then, we show the results of our method and some other baseline methods on the benchmark. Lastly, we provide an ablation study and robustness test of our network. We evaluate how our model works both quantitatively and qualitatively.

4.1. Data Generation and Model Training

We tested our method on a commonly used benchmark for 3D point cloud completion, i.e., ShapeNet. We used 13 and 16 categories of different objects in the benchmark dataset ShapeNet-Part [27]. Since a more diverse dataset can comprehensively test the capabilities of the model, we also conducted experiments using all 55 categories in ShapeNet. Similar to the dataset of PoinTr, we randomly sampled 80% of the objects from each category of ShapeNet-55 to form the training set, and 20% for evaluation, resulting in 41,952 models for training and 10,518 models for testing. Complete point clouds were generated by uniformly sampling 2048 points from 5 randomly distributed viewpoints. During training and testing, we sampled 512 points as the ground truth F_{gt} of the incomplete point cloud, that is, 25% of the original data were missing. We applied iterative farthest point sampling (IFPS) [28] to extract ground truths F'_{gt} and F''_{gt} with 128 and 64 points, respectively.

We implemented all of our models with the PyTorch deep learning framework, and used the Adam optimizer to update the parameters of the network during training. The learning rate was initially set to 0.0001, the model was trained for 200 epochs, and the continuous learning rate decreased by 0.2 every 40 epochs. We defined the batch size to be 24. The models were trained and tested on a NVIDIA GeForce RTX 3090 GPU with CUDA 11.7. The mixed attention module can be used flexibly in convolutional neural networks (CNNs) to capture feature correlations effectively, and the bidirectional point pyramid attention network (BiPPAN) can be used as a decoder in the network.

4.2. Completion Results on ShapeNet

We applied the mean Chamfer Distance as the evaluation metric, which contained two items in our experiments: Pred $\rightarrow$ GT and GT $\rightarrow$ Pred.

The Chamfer Distance from a predicted point cloud to the ground truth (Pred $\rightarrow$ GT) measures the difference between the predicted and real point clouds, which is calculated with:

$$d_{CD}(F_{Pred}, F_{GT}) = \frac{1}{F_{Pred}} \sum_{x \in F_{Pred}} \min_{y \in F_{GT}} \|x - y\|_2^2 \quad (8)$$

The Chamfer Distance from the ground truth to a predicted point cloud (GT $\rightarrow$ Pred) manifests the extent to which the shape of the prediction covers the real point cloud, which is calculated with:

$$d_{CD}(F_{GT}, F_{Pred}) = \frac{1}{F_{GT}} \sum_{x \in F_{GT}} \min_{y \in F_{Pred}} \|x - y\|_2^2 \quad (9)$$

Results on ShapeNet-13: We compared the results of our model with those of L-GAN [29], PCN [15], 3D-point Capsule Networks [30] and PF-Net [14] on standard metrics. In Table 1, the results of the above four methods are cited from [14]. Our method surpassed the other ones on 13 categories in terms of Pred $\rightarrow$ GT and GT $\rightarrow$ Pred errors, and achieved the best average Chamfer Distance in the missing region. This suggests that our network was able to reconstruct the missing point cloud with higher precision.

Results on ShapeNet-16: We experimented with our method and the most advanced PF-Net method on ShapeNet-16. Since PF-Net was only trained on the ShapeNet-13 dataset, we trained and tested it again using the same dataset according to their open-source code for fair comparison and quantitative analysis. Table 2 shows that our method resulted in a lower average Chamfer Distance error when increasing the dataset from 13 to 16 categories. The three new categories, namely, Earphone, Knife, Rocket, on both Pred $\rightarrow$ GT and

GT → Pred errors were all smaller than those of PF-Net. These results justify that the proposed network could handle a larger dataset and effectively reconstruct the full shape. Figure 6 shows the qualitative results of our method and PF-Net on ShapeNet-16. Our method preserved the original input geometry while better computing and refining the missing parts of the completion for the shown examples.

Table 1. Results of a comparison between our method and most advanced methods on the ShapeNet-13 dataset. The numbers in pairs are Pred → GT and GT → Pred Chamfer Distance × 10^3 (lower is better).

Category	LGAN-AE	PCN	3D Capsule	PF-Net	BiPPAN
Airplane	3.357/1.130	5.060/1.243	2.676/1.401	1.091/1.070	0.964/0.875
Bag	5.707/5.303	3.251/4.314	5.228/4.202	3.929/3.768	2.864/2.823
Cap	8.968/4.608	7.015/4.240	11.04/4.739	5.290/4.800	4.371/3.489
Car	4.531/2.518	2.741/2.123	5.944/3.508	2.489/1.839	2.28/1.681
Chair	7.359/2.339	3.952/2.301	3.049/2.207	2.074/1.824	1.758/1.442
Guitar	0.838/0.536	1.419/0.689	0.625/0.662	0.456/0.429	0.377/0.376
Lamp	8.464/3.627	11.61/7.139	9.912/5.847	5.122/3.460	3.731/2.513
Laptop	7.649/1.413	3.070/1.422	2.129/1.733	1.247/0.997	1.118/0.864
Motorbike	4.914/2.036	4.962/1.922	8.617/2.708	2.206/1.775	1.931/1.562
Mug	6.139/4.735	3.590/3.591	5.155/5.168	3.138/3.238	3.036/2.914
Pistol	3.944/1.424	4.484/1.414	5.980/1.782	1.122/1.055	0.929/0.831
Skateboard	5.613/1.683	3.025/1.740	11.49/2.044	1.136/1.337	1.007/1.029
Table	2.658/2.484	2.503/2.452	3.929/3.098	2.235/1.934	1.744/1.626
Mean	5.395/2.603	4.360/2.661	5.829/3.008	2.426/2.117	2.008/1.694

Table 2. Results of a comparison between our method and PF-Net on ShapeNet-16 dataset. We also provide the number of parameter (Params) in the last rows.

Category	PF-Net		BiPPAN	
	Pred → GT	GT → Pred	Pred → GT	GT → Pred
Airplane	1.084	1.119	0.963	0.872
Bag	3.979	4.668	2.83	2.624
Cap	5.254	4.897	4.004	3.614
Car	2.548	1.914	2.316	1.634
Chair	2.154	2.019	1.738	1.428
Earphone	6.003	8.058	3.194	3.690
Guitar	0.464	0.546	0.377	0.382
Knife	0.555	0.563	0.464	0.465
Lamp	4.943	3.883	3.863	2.557
Laptop	1.309	1.072	1.135	0.888
Motorbike	2.328	1.836	1.944	1.581
Mug	3.080	3.580	3.147	2.893
Pistol	1.284	1.053	0.920	0.811
Rocket	1.052	0.762	0.853	0.620
Skateboard	1.196	1.362	1.025	1.004
Table	2.305	2.123	1.760	1.629
Mean	2.471	2.466	1.908	1.668
Params ($\times 10^6$)	76.571		11.763	

Furthermore, we compared the number of trainable parameters in PF-Net and our method. As shown in Table 2, our method performed significantly better with fewer parameters than those of PF-Net.



Figure 6. Qualitative results on ShapeNet-16 showing the input point cloud (Input) and the ground truth (G.T.), and the predictions of PF-Net and our model.

Results on ShapeNet-55: To further investigate how PF-Net and our method performed with more dataset categories, we conducted experiments on ShapeNet-55. Due to the imbalanced number of shapes in the dataset, we classified the 55 categories into simple, moderate, and hard classes on the basis of their number. Specifically, more than 2500 training shapes were considered to be of the simple class, such as table, chair, and airplane. Fewer than 80 samples were classified as of the hard class, such as birdhouse, bag, and keyboard. Bed, camera, and rifle were classified as of the moderate class. We chose 3 categories from each class as examples to demonstrate the results. The average CD results for the three classes are shown in Table 3 with details. The category errors and average errors of our method were smaller than those of PF-Net. The results of our method on ShapeNet-55 were especially better than those on ShapeNet-13 and ShapeNet-16. Due to the imbalanced number of training samples, the average CD errors of the moderate and hard classes with insufficient samples for PF-Net were higher than those of the simple class. For our BiPPAN, there were no significant differences in the average errors of the three classes. In Figure 7, we present the qualitative results for 9 categories on Shapenet-55. As shown in the example, our method was able to complete the missing point cloud with higher accuracy and more details for the various incomplete categories.

Regardless of quantitative comparison or qualitative analysis, our method could improve the quality of the completed point cloud on a diverse dataset and achieve good results in the point cloud completion task.

Table 3. Results of a comparison between our method and PF-Net on the ShapeNet-55 dataset. The numbers in pairs are Pred → GT and GT → Pred Chamfer Distance × 10^3 (lower is better). CD-S, CD-M, and CD-H were used to represent CD errors in the simple, moderate, and hard classes.

	Simple			Moderate			Hard			CD-S	CD-M	CD-H	CD-Ave
	Table	Chair	Airplane	Bed	Camera	Rifle	Birdhouse	Bag	Keyboard				
PF-Net	1.963 /1.882	2.101 /1.975	1.162 /1.414	3.327 /4.742	4.827 /5.613	1.235 /1.029	3.689 /3.650	3.896 /3.107	1.911 /1.006	1.845 /1.737	2.464 /2.142	2.744 /2.362	2.463 /2.150
BiPPAN	1.386 /1.156	1.591 /1.177	0.884 /0.874	2.741 /1.987	3.258 /2.321	0.541 /0.488	2.015 /2.770	1.889 /1.289	0.766 /0.594	1.449 /1.134	1.801 /1.301	1.647 /1.245	1.738 /1.274

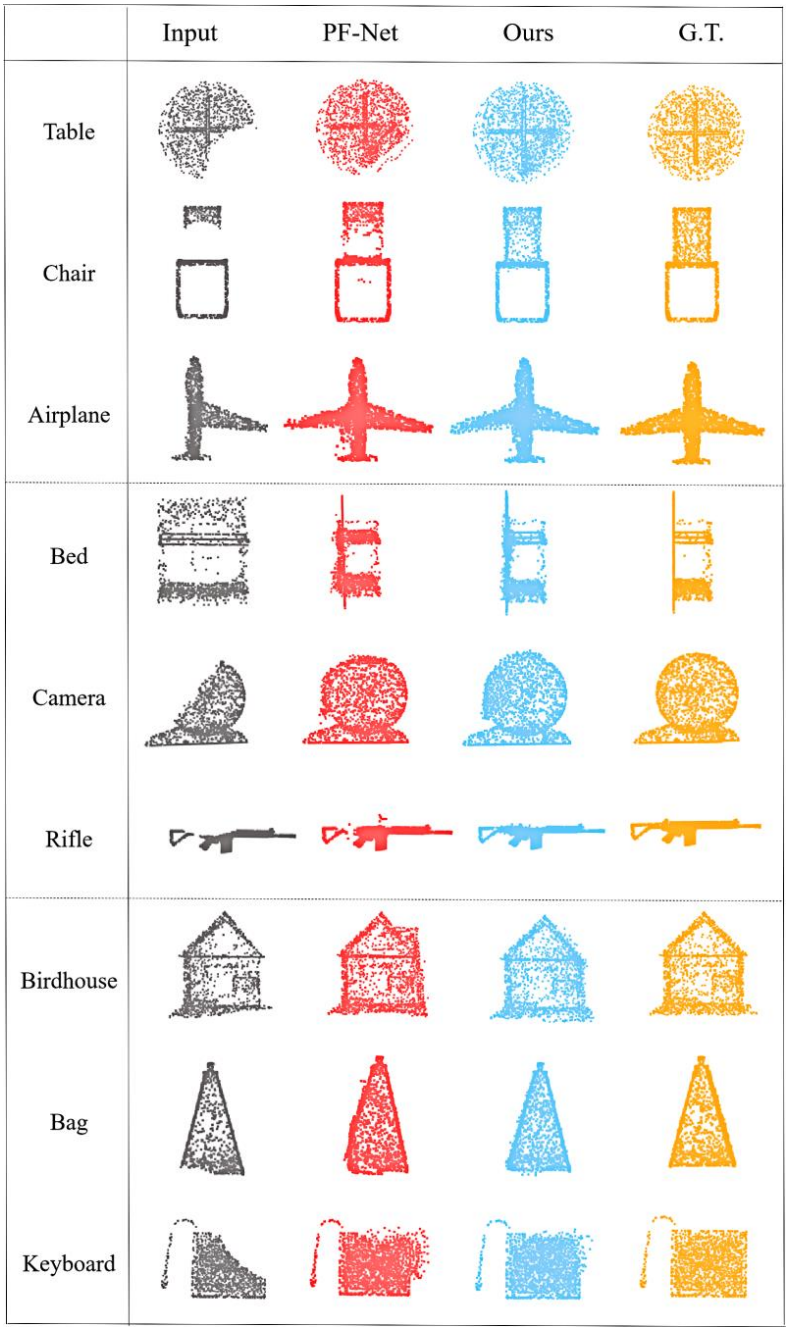


Figure 7. Qualitative results on ShapeNet-55. Point cloud completion results of our method and PF-Net on some objects from the simple, moderate, and hard classes.

4.3. Ablation Study

To test the effectiveness and necessity of the three designed modules of the Attention Mechanism (AM, including the Channelwise Attention Module of the encoder and the Mixed Attention Module of the decoder), the Bidirectional Point Pyramid Network (BiPPN), and skip connections (Skip-C), we exhaustively compared and tested our method BiPPAN with its four variants on ShapeNet-16. The results of the ablation study are shown in Table 4.

Table 4. Ablation study on different components of our proposed encoder–decoder network framework, including Attention Module (AM), Bidirectional Point Pyramid Network (BiPPN), and skip connections (Skip-C).

Model	AM	BiPPN	Skip-C	Pred → GT ($\times 10^3$)	GT → Pred ($\times 10^3$)
A				2.498	2.300
B		√	√	2.125	1.869
C	√		√	2.242	1.837
D	√	√		2.004	2.011
BiPPAN	√	√	√	1.908	1.668

Model A is a very basic baseline that includes the standard CMLP and FPN as the encoder and decoder architectures, respectively. Models B, C, D had the same structure as that of BiPPAN (our entire network) except for the removed/replaced module. Model B was the variant where AM was removed from the BiPPAN model. Adding AM reduced Pred → GT and GT → Pred errors. This confirmed the effectiveness of AM. Model C focused on demonstrating the effectiveness of introducing BiPPN into network design. After changing the decoder from FPN into BiPPN, Pred → GT and GT → Pred errors were smaller, as expected. When removing Skip-C from the BiPPAN model, the GT → Pred error of Model D increased markedly. This clearly shows that Skip-C alone can bring significant performance improvement. Notably, our BiPPAN method performed better than its variants. A visual comparison is presented in Figure 8. For the shown examples, Baseline Model A and Models B, C, D suffered from some obvious fairness coarsening and exhibited impotence for most of the details. This further confirmed the effectiveness of the BiPPAN design.

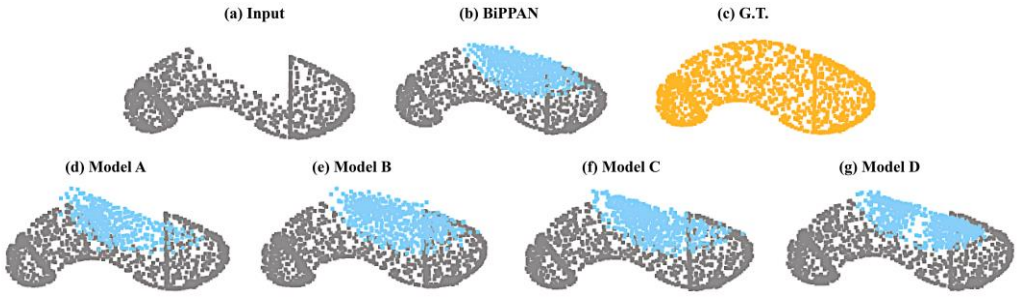


Figure 8. Comparison of point cloud completion results of the proposed network, baseline Model A, and Models B, C, D.

4.4. Robustness Test

In order to analyze the robustness of our method, we evaluated the point cloud completion of missing regions in different degrees on ShapeNet-16. We varied the number of incomplete point clouds from 512 to 1024 and 1536. In other words, we extended the missing part from 25% to 50% and 75% of the original shape. Table 5 and Figure 9 show the experimental results of the network on the Airplane and Car categories. Table 5 shows that the errors between the predicted point cloud and the ground truth did not differ much when 25%, 50%, and 75% of the points were missing. Figure 9 displays that, although the

partial data were quite tough, our predictions were sensible. In the case of the 75% missing area, the complete car shape could be defined and completed only on the basis of the front of the car. This experiment verified the strong completion robustness of our method when dealing with input point clouds with distinct missing degrees.

Table 5. Robustness results of different extent of incomplete input point cloud using our method.

Missing Ratio	**25%**		**50%**		**75%**	
CD ($\times 10^3$)	Pred → GT	GT → Pred	Pred → GT	GT → Pred	Pred → GT	GT → Pred
Airplane	0.963	0.872	0.971	0.809	0.968	0.896
Car	2.316	1.634	2.494	1.786	2.479	1.841

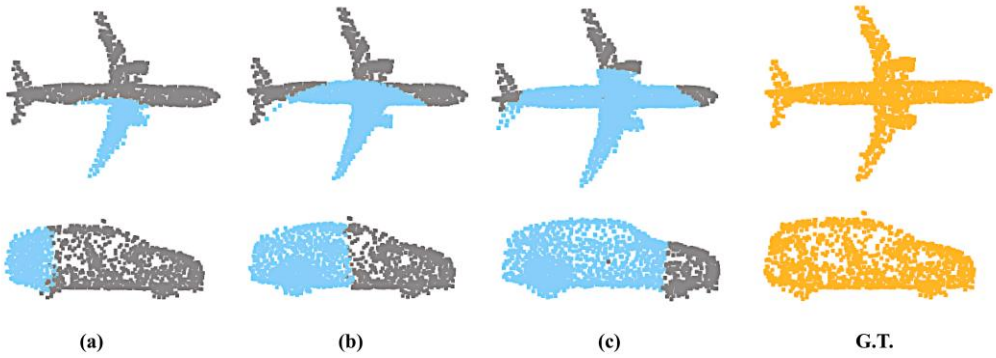


Figure 9. Examples of point cloud completion results when the input point cloud misses (**a**) 25%, (**b**) 50%, and (**c**) 75% of the original point cloud. Gray and blue denote the input incomplete point cloud and predicted missing point cloud, respectively. Yellow represents the real point cloud.

5. Conclusions

In this paper, we recommended a novel Bidirectional Point Pyramid Attention Network framework (BiPPAN) for point cloud completion. Our method produces missing point clouds via an encoder–decoder structure. The attention module was introduced into the encoder and decoder to raise the correlational information of point clouds, which extracts the local details of object features better [31]. The point pyramid attention decoder was designed not only to fully use hierarchical features from the bidirectional connection, but also to capture long-distance structural information by employing skip connections as well [32]. Experimental results on ShapeNet-13, ShapeNet-16, and ShapeNet-55 demonstrate that, no matter the size of the dataset, our proposed method was effective and efficient in comparison with other methods. An ablation and robustness test validated our claim that the attention module, bidirectional connection, and skip connections could more effectively extract and exploit features of point clouds to improve prediction accuracy. Our method tended to reduce the error in challenging classes.

Hence, this method can be widely applied to the task of generating target point clouds with both rich semantic profiles and detailed features while preserving existing contours. The good results of BiPPAN on many datasets demonstrate its ability to repair and complete 3D shapes, and improve the accuracy of downstream tasks such as 3D recognition, and 3D object detection and tracking. As a point cloud preprocessing method, our method has tremendous potential in autonomous vehicles, 3D reconstruction, and remote sensing research.

Author Contributions: Methodology and writing—original draft preparation, Y.L.; data curation and visualization, Y.X.; investigation and supervision, J.G.; conceptualization and project administration, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Young Science and Technology Star Project of Dalian, China [grant number 2022RQ092], and the Technology Innovation Fund Project of Dalian Neusoft University of Information [grant number TIFP202303].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The datasets can be found here: https://www.shapenet.org/ (accessed on 17 November 2022).

Acknowledgments: The authors acknowledge all editors and reviewers for their suggestions. We would also like to thank the Dalian Ascend AI Computing Center and Dalian Ascend AI Ecosystem Innovation Center for providing computing power and technical support.

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

References

1. Wen, X.; Han, Z.; Cao, Y.-P.; Wan, P.; Zheng, W.; Liu, Y. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13080–13089.
2. Sun, Y.; Wang, Y.; Liu, Z.; Siegel, J.E.; Sarma, S.E. PointGrow: Autoregressively Learned Point Cloud Generation with Self-Attention. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020.
3. Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; Su, J.K. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*; NeurIPS: San Diego, CA, USA, 2019; Volume 32.
4. Santhanavijayan, A.; Kumar, D.N.; Deepak, G. A semantic-aware strategy for automatic speech recognition incorporating deep learning models. In *Intelligent System Design*; Springer: Singapore, 2021; pp. 247–254.
5. Pandey, B.; Pandey, D.K.; Mishra, B.P.; Rhmann, W. A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: Challenges and research directions. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *34*, 5083–5099. [CrossRef]
6. Zhang, W.; Li, H.; Li, Y.; Liu, H.; Chen, Y.; Ding, X. Application of deep learning algorithms in geotechnical engineering: A short critical review. *Artif. Intell. Rev.* **2021**, *54*, 5633–5673. [CrossRef]
7. 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/CVF Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
8. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *Comput. Sci.* **2015**. [CrossRef]
9. Cui, Y.; Chen, R.; Chu, W.; Chen, L.; Tian, D.; Li, Y.; Cao, D. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 722–739. [CrossRef]
10. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway Township, NJ, USA, 2020; Volume 43, pp. 4338–4364.
11. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
12. Alliegro, A.; Valsesia, D.; Fracastoro, G.; Magli, E.; Tommasi, T. Denoise and contrast for category agnostic shape completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4629–4638.
13. 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, 10–17 October 2021; pp. 5499–5509.
14. 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.
15. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. In Proceedings of the International Conference on 3D Vision, Verona, Italy, 5–8 September 2018.
16. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
17. Hu, T.; Han, Z.; Shrivastava, A.; Zwicker, M. Render4Completion: Synthesizing Multi-View Depth Maps for 3D Shape Completion. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, Seoul, Korea, 27–28 October 2019.

18. Huang, T.; Zou, H.; Cui, J.; Yang, X.; Wang, M.; Zhao, X.; Zhang, J.; Yuan, Y.; Xu, Y.; Liu, Y. Rfnet: Recurrent forward network for dense point cloud completion. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 12508–12517.
19. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
20. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
21. Miculicich, L.; Ram, D.; Pappas, N.; Henderson, J. Document-level neural machine translation with hierarchical attention networks. In *Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018.
22. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
23. Wen, X.; Li, T.; Han, Z.; Liu, Y.S. Point Cloud Completion by Skip-attention Network with Hierarchical Folding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
24. 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 Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 12498–12507.
25. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway Township, NJ, USA, 2018.
26. Fan, H.; Su, H.; Guibas, L. 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.
27. Yi, L.; Kim, V.G.; Ceylan, D.; Shen, I.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; Guibas, L. A scalable active framework for region annotation in 3D shape collections. *ACM Trans. Graph.* **2016**, *35*, 1–12. [CrossRef]
28. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
29. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning Representations and Generative Models for 3D Point Clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
30. Zhao, Y.; Birdal, T.; Deng, H.; Tombari, F. 3D Point Capsule Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
31. Wang, J.; Cui, Y.; Guo, D.; Li, J.; Liu, Q.; Shen, C. Pointattn: You only need attention for point cloud completion. *arXiv* **2022**, arXiv:2203.08485.
32. Bai, Y.; Wang, X.; Ang, M.H., Jr.; Rus, D. BIMS-PU: Bi-Directional and Multi-Scale Point Cloud Upsampling. In *IEEE Robotics and Automation Letters*; IEEE: Piscataway Township, NJ, USA, 2022; pp. 7447–7454.

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.