Multi-Scale Feature Optimization Point Cloud Completion Network Integrating SoftPool

Wanpeng Zhang and Ziwei Zhou*

Abstract—**Point cloud completion is crucial in point cloud processing, as it can repair and refine incomplete 3D data, ensuring more accurate models. However, current point cloud completion methods commonly face a challenge: they fail to fully utilize multi-scale information from local features, leading to limitations in accuracy and detail preservation. To address this issue, this paper proposes a multi-scale feature optimization algorithm for point cloud completion that integrates SoftPool.Based on DGCNN, the method combines dilated convolution and bottleneck attention mechanisms to extract features at different scales, enhancing the ability to capture detailed information in point clouds. The bottleneck attention mechanism is used to optimize important detail features. The extracted local features are concatenated with their corresponding positional information to form point proxies, enhancing the effective extraction of local geometric features, resulting in more refined completed point cloud shapes. A Transformer architecture is employed to model these features.Finally, SoftPool is introduced for fine-grained feature downsampling, improving the network's ability to recover point cloud details. FoldingNet is used to reconstruct missing structures and output the completed point cloud. To validate the model's completion performance, training and testing are conducted on the PCN and ShapeNet55 datasets. Experimental results demonstrate that the model has better feature detail retention and more accurate completion results. On the PCN dataset, the average CD value is reduced by 6.5% compared to the best-performing model among the comparison methods. On the ShapeNet55 dataset, the average CD value across three difficulty levels is reduced by 6.9% compared to the best-performing model among the comparison methods. Additionally, the model also achieved a 2.1% improvement in F-score.**

Index Terms—**Multi-Scale Feature, Point Cloud Completion, SoftPool, Transformer, Bottleneck Attention**

I. Introduction

WITH the rapid development of 3D vision technology, depth sensors such as LiDAR and RGB-D cameras have been widely applied in fields like autonomous driving and robotics, enabling efficient acquisition of 3D point cloud data. However, factors such as device accuracy, acquisition angles, occlusion, reflection, transparency, and lighting conditions often result in sparse and incomplete point cloud data, which negatively impacts the accuracy and reliability of downstream tasks. Point cloud completion plays a fundamental role in complex tasks such as high-resolution 3D map reconstruction and underground mining environment reconstruction. And it is crucial for the performance of 3D object detection and 3D shape classification. Therefore, conducting in-depth research on point cloud completion is of great significance.

In recent years, with the development of deep learning, point cloud completion methods have made continuous progress, gradually shifting from traditional geometric inference and model alignment methods to deep learning-based techniques. These methods achieve high-quality completion of sparse and incomplete point clouds by extracting global and local features from large-scale data to establish mapping relationships between incomplete and complete point clouds. PCN [1] is an early deep learning-based point cloud completion network. Its encoder consists of multiple PointNet [2] units, and its decoder divides the point cloud generation task into two stages: coarse generation and fine generation. In the fine stage, FoldingNet [3] is used to generate denser point cloud predictions, effectively completing the missing point clouds. However, PCN's feature extraction module relies on multilayer perceptrons to process point clouds, which neglects the local information of incomplete point clouds, resulting in poor completion of local details and difficulty in generating high-fidelity point cloud results. Methods such as PoinTr [4] and PF-Net [5] have achieved significant results in point cloud completion, but these methods typically only address local incompleteness and fail to consider geometric shape loss and sparsity simultaneously when generating complete point clouds or filling missing areas. Consequently, these methods do not perform well in cases where the point cloud is sparse. Furthermore, current Transformer-based point cloud completion networks perform well but mostly use max pooling for downsampling. Max pooling keeps only the most prominent features, often causing a loss of many detailed features.

In summary, many existing point cloud completion methods extract only single features from the input point cloud, failing to fully exploit the intrinsic multi-level structure and semantic information, leading to limitations in accuracy and detail preservation. To improve the effectiveness of point cloud completion, a method capable of capturing rich features and semantic information is needed to achieve more accurate and detailed completion results. Therefore, this paper proposes a Multi-Scale Feature

Manuscript received August 29, 2024; revised November 23, 2024.

This work was supported by the Natural Science Foundation of China(Grant No. 61575090), the Natural Science Foundation of China Youth Fund(No. 61803189), Natural Science Foundation of Liaoning Province(2019-ZD-0031 and 2020FWDF13).

Wanpeng Zhang is a postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (phone:86-16642299371, e-mail: 584593254@qq.com).

Ziwei Zhou* is an Associate Professor at the School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan 114051, China (Corresponding author to provide phone: 86-139-4125-5680; e-mail: 381431970@qq.com).

Optimization Point Cloud Completion Network with SoftPool integration, abbreviated as MFOSNet. The Multi-Scale Feature Optimization (MSFO) module is designed based on the DGCNN architecture, combining dilated convolution [6] and the Bottleneck Attention Module (BAM) [7]. By extracting and integrating features at different scales, the method enhances the ability to capture detailed information in the point cloud, and the bottleneck attention mechanism is used to enhance important detail features, thereby improving the detail preservation of the completion results. During the downsampling stage, the SoftPool pooling method is used to ensure that the features of each element influence the downsampling result. This allows for more detailed feature downsampling without losing important details, thereby enhancing the network's ability to restore point cloud details.

II. Related Jobs

Geometry-based methods, including the approach used by Thrun et al. in 2005 [8] to reconstruct occluded surfaces by leveraging object symmetry, the method proposed by Schnabel et al. in 2009 [9] that completes missing regions by extending the surrounding structures, and the arterial snake model introduced by Li et al. in 2010 [10] which combines topological and geometrical information to fill large missing areas, have been developed for completion tasks. While these methods are computationally simple and efficient, they perform poorly in cases of large missing areas or asymmetry, limiting their practical applicability.

Template matching-based methods achieve completion by retrieving the most similar template point cloud from a database. In 1999, Blanz et al. [13] deformed the retrieved model to synthesize a consistent shape. In 2005, Pauly et al. [11] used non-rigid alignment methods to complete missing parts. In 2014, Yin et al. [14] employed geometric primitives instead of a shape database for repair, while in 2015, Sung et al. [12] retrieved the best-fitting parts through part alignment. These methods yield good completion results, but the retrieval process from large databases consumes significant computational resources, making real-time processing difficult. The optimization process is particularly expensive in the presence of noise.

Specifically, geometric methods and template matching methods each have their own advantages and disadvantages. Geometric methods are computationally simple but perform poorly when dealing with large missing areas or asymmetrical structures. Template matching methods yield better completion results, but they require significant computational resources, making real-time processing difficult. For practical applications, it is essential to find a balance between accuracy and computational efficiency. The rise of deep learning methods has brought new breakthroughs. In 2017, Wang et al. [15] proposed a hybrid framework to generate 3D models with semantic coherence and contextual details. This framework combines a 3D Encoder-Decoder Generative Adversarial Network (3D-ED-GAN) with a Long-term Recurrent Convolutional Network (LRCN). The 3D-ED-GAN is responsible for filling in missing 3D data at low resolution, while the LRCN is used to localize fine-grained details. However, voxel-based techniques cannot achieve fine detail in 3D reconstruction because as resolution increases, the network's complexity and computational requirements grow sharply, limiting the resolution of the reconstructed voxels due to memory and computational constraints. In 2018, Achlioptas et al. [16] were the first to apply deep learning to point cloud completion, proposing the LGAN-AE network model, which effectively completes point clouds using a Generative Adversarial Network (GAN). However, its decoder struggles to recover rare geometric structures.

In recent years, new methods have continuously emerged. In 2020, Xie et al. [17] proposed GRNet, which converts unordered point cloud data into a regular voxel grid, generating predicted point clouds through 3D convolutional layers. In 2021, Wang et al. [18] introduced a point cloud completion method based on style and adversarial differentiable rendering. In 2023, Ma et al. [19] proposed the MFCPNet network, which progressively integrates features at different scales and utilizes the Transformer's Encoder-Decoder structure to generate missing point clouds, further improving completion results. However, the aforementioned methods still have limitations. To address these issues, a Multi-Scale Feature Optimization Point Cloud Completion Network Integrating SoftPool is proposed. This approach aims to overcome the current models' inability to deeply explore complex structures and semantic information within point clouds, which has led to limitations in the accuracy and detail preservation of completion results, thereby affecting the overall effectiveness of point cloud completion.

III. Model Design

The structure of the Multi-Scale Feature Optimization Point Cloud Completion Network with SoftPool integration is shown in Figure 1. The overall network architecture consists of two main parts. The first part involves feature extraction and point proxy generation. Initially, the Farthest Point Sampling (FPS) method is used to extract central points from the incomplete point cloud to ensure representative coverage of the entire point cloud. Subsequently, the Multi-Scale Feature Optimization (MSFO) module is employed to extract local features around the central points, capturing the geometric information and local structures of the point cloud. To further enhance the expressiveness of the features, a simple Multilayer Perceptron (MLP) is used to extract positional embeddings for each local feature, which are then fused with the local features extracted by the MSFO, ultimately generating the point proxies. The second part focuses on the fine reconstruction of the point cloud. The point proxies are fed into the Transformer encoder for global feature modeling. The output of the query generator is then passed to the decoder to generate the predicted point proxies. Subsequently, SoftPool is used for downsampling, which reduces redundant data while retaining critical features. Finally, the FoldingNet network is employed to progressively refine the point cloud structure through a series of folding operations, ensuring that the final output is a complete and detailed point cloud.

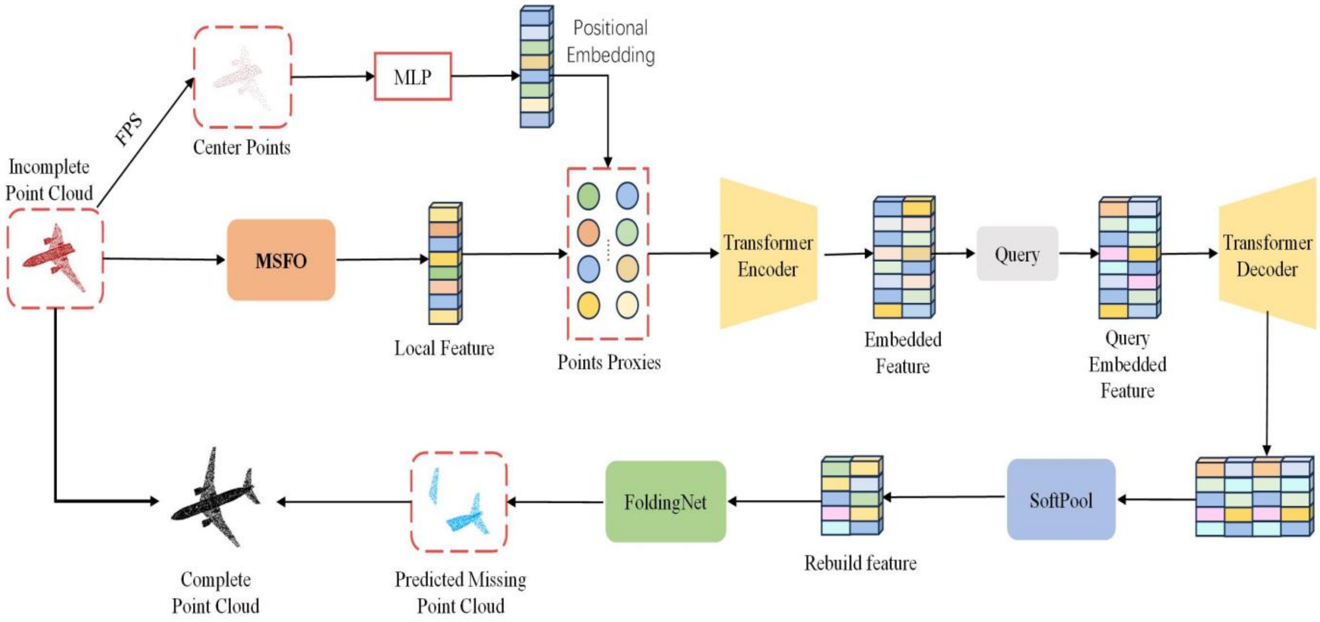

Fig. 1. The overall structural diagram of the Multi-Scale Feature Optimization Point Cloud Completion Network Integrating SoftPool

A. *Multi-Scale Feature Optimization Module*

The standard DGCNN network [20] employs EdgeConv to extract local geometric features from point cloud data. However, EdgeConv is constrained to Single-Scale Feature extraction, meaning it aggregates features solely within a fixed neighborhood. This limitation hinders its ability to capture Multi-Scale spatial information, resulting in inadequate detail representation—particularly for critical features in incomplete point clouds where essential details may be absent or obscured. To address this challenge, this paper introduces the Multi-Scale Feature Optimization (MSFO) module. By implementing a Multi-Scale Feature extraction mechanism, MSFO effectively captures both local and global features across various scales. This approach not only enhances the feature representation of point clouds but also preserves and refines crucial detail features in incomplete point clouds, thereby augmenting the network’s overall capability to interpret point cloud data. This module successfully mitigates the limitations inherent in traditional EdgeConv by emphasizing critical detailed features. The architecture and workflow of the MSFO module are illustrated in Figure 2.

Inspired by the set abstraction mechanism in PointNet++ [21], we first select several center points in the incomplete point cloud using the Farthest Point Sampling method. A feature transformer is then used to expand the feature dimension of the center points from 3D to 8D, enriching the input features for subsequent network layers. The features are then fed into multiple EEC modules. Finally, the refined local region features are obtained through the bottleneck attention module.

The EEC module integrates dilated convolution into EdgeConv to extract features at different scales. First, using the KNN algorithm [22], the K nearest points $\left(g_{i_1}, g_{i_2}, \ldots g_{i_k}\right)$ are selected around the point g_i as the center. The feature of each point g_i is denoted as x_i , and the features of its K nearest points are denoted as s_k . The difference features $\left[s_k - x_i\right]$ between point g_i and the K points are calculated and concatenated with the features of point g_i to construct the local neighborhood feature F_{edg} , thereby capturing the local geometric information in the point cloud. The formula is:

$$F_{edg} = x_i \oplus \left(s_k - x_i\right) \quad (1)$$

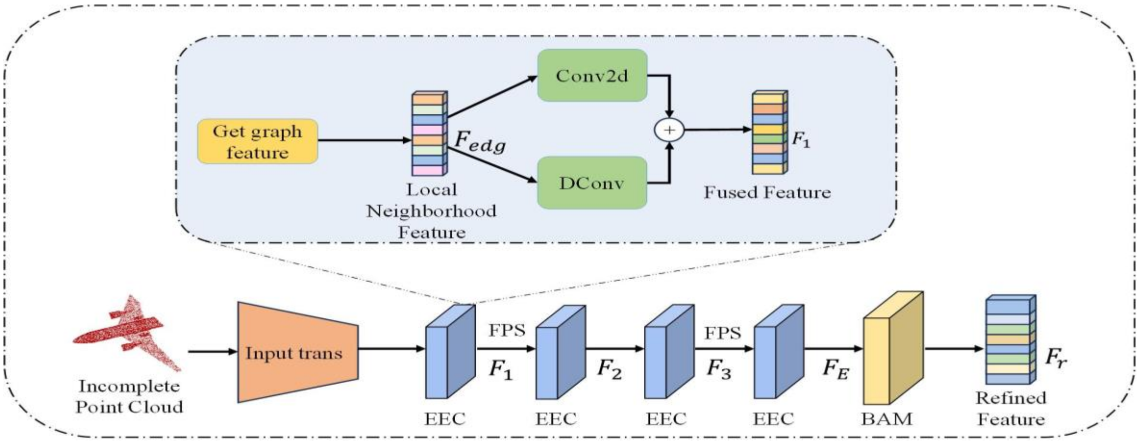

Fig. 2. The structure diagram of the Multi-Scale Feature Optimization Module

The local neighborhood features are fed into both 2D convolution and dilated convolution for multi-scale feature extraction. Group normalization and $LeakyReLU$ activation functions are applied to the 2D convolution layers for non-linear transformation. Dilated convolution with a dilation rate of 2 is used to extract features over a larger receptive field, allowing for the capture of different details within the point cloud. Finally, the features from the two scales are concatenated to obtain the fused local region features. The formula is:

$$F_1 = LReLU\left(Group\left(Conv2d\left(F_{edg}\right)\right)\right) \oplus DConv\left(F_{edg}\right) \quad (2)$$

Where $LRelu$ represents the $LeakyReLU$ activation function, $Group$ stands for the group normalization layer, $Conv2D$ denotes the 2D convolution, $DConv$ represents the dilated convolution, and $\oplus$ indicates the concatenation operation.

The local regional features F_1 obtained from the first layer of the EEC module are sequentially passed through the other three layers of EEC modules, with two FPS operations in between, ultimately resulting in the local regional features F_E.

The local region features F_E are fed into the Bottleneck Attention Module (BAM), which employs parallel spatial and channel attention mechanisms to enhance critical local features, as illustrated in Figure 3. The channel attention mechanism adjusts the channel weights to emphasize key detailed features while suppressing redundant information, thereby enhancing the feature representation capabilities. Meanwhile, spatial attention assigns varying attention weights across the spatial dimension, facilitating a better understanding of local structures and enabling the extraction of more precise point cloud details. Consequently, this improves the representation of point cloud features. The final refined local region features are obtained, and the formula is:

$$F_r = F + F \otimes \sigma\left(M_c\left(F\right) + M_s\left(F\right)\right) \quad (3)$$

Where F_r represents the refined local region features, σ is the sigmoid function, $M_C(F)$ and $M_S(F)$ represent the channel attention map and spatial attention map, respectively, and $\otimes$ denotes element-wise multiplication. The expressions for $M_C(F)$ and $M_S(F)$ are as follows:

$$M_c\left(F\right) = BN\left(MLP\left(AvgPool\left(F\right)\right)\right) \quad (4)$$

$$M_s\left(F\right) = BN\left(f_3^{1\times1}\left(f_2^{3\times3}\left(f_1^{3\times3}\left(f_0^{1\times1}\right)\right)\right)\right) \quad (5)$$

Where BN represents the batch normalization layer, MLP stands for the multilayer perceptron, $AvgPool$ denotes global average pooling, and f refers to the convolution operation, with the superscript indicating the size of the convolution kernel.

B. Point Proxy Generation

Adopting the Transformer architecture for point cloud completion tasks takes full advantage of its ability to capture global information, position invariance, scalability, parallel processing capabilities, and robust modeling power. By leveraging these strengths, the Transformer effectively establishes connections between global contextual information and local details, enabling efficient and precise point cloud completion. Point proxies play a critical role in providing the Transformer with structured, dense, and fixed-size input representations, addressing challenges such as data sparsity, irregularity, and computational complexity. This design ultimately enhances the model's efficiency and overall performance.

Define the point cloud as a series of "point proxies," with each point proxy representing the features of a local region within the point cloud. The position embedding for each local region is extracted using an MLP network. Finally, the point proxy is obtained by adding the position information to the local region features F_r derived from the MSFO module. The formula for the point proxy is given as:

$$F_r' = F_r + \varphi\left(g_i\right) \quad (6)$$

The term F_r represents the local region feature, centered at g_i and extracted by the MSFO module, which encapsulates the semantic information of each point’s local region. The parameter φ denotes the MLP used to capture the position information of the point proxy.

This representation effectively integrates local features with spatial position information, enabling the model to capture essential details in point clouds at both global and local levels. By leveraging point proxy design, the Transformer significantly improves its ability to process point cloud data, excelling in the completion of complex shapes and sparse point clouds.

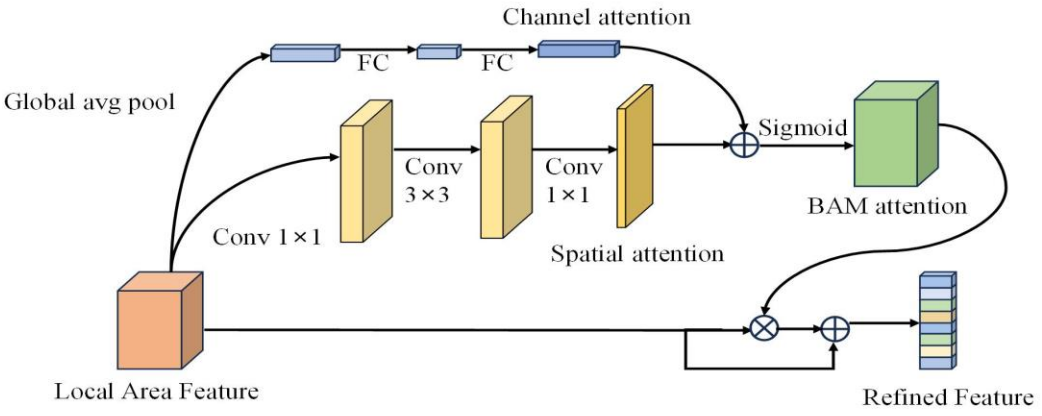

Fig. 3. The structure diagram of the Bottleneck Attention Module

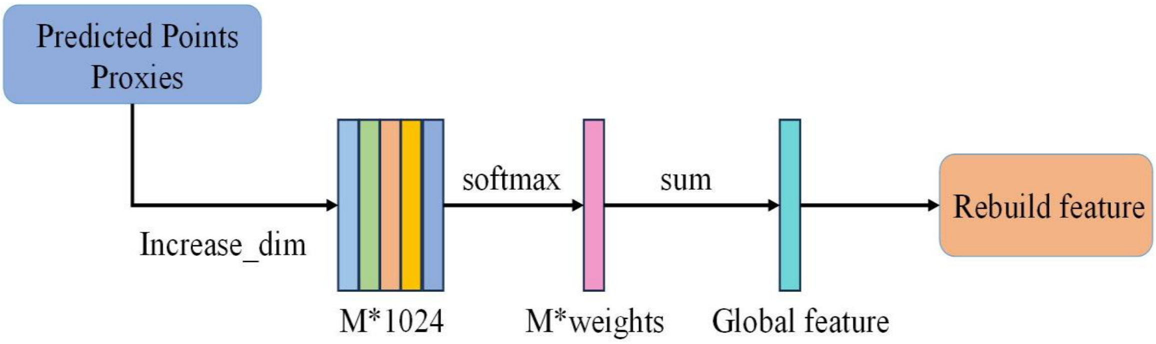

Fig. 4. SoftPool Diagram

C. *Fine Reconstruction of the Point Cloud*

The obtained point proxies are used as input to the Transformer encoder for global feature modeling. The output from the query generator is then passed to the decoder to generate the predicted point proxies. The expression is as follows:

$$P = T_D(Q,V) \tag{7}$$

The symbols are defined as follows: Q represents the dynamic queries, V denotes the feature vectors output by the encoder, TD refers to the decoder, and $P = \{P_1, P_2, \ldots P_M\}$ represents the predicted point proxies, where M indicates the number of predicted points.

Most point cloud completion networks typically use max pooling or average pooling methods to downsample point cloud data before feeding the features into FoldingNet. Max pooling reduces data dimensionality by selecting the maximum value within the pooling region, maintaining data permutation invariance, while average pooling achieves feature downsampling by calculating the average value of all elements. However, max pooling retains only the maximum value from the region's features, leading to a significant loss of detailed information, and average pooling, which averages all elements within the region, also results in the loss of many critical details. Therefore, this paper adopts the Softpool pooling method to process the input feature vectors, as illustrated in Figure 4. Compared to max pooling and average pooling, Softpool better preserves the feature information during the downsampling process, allowing for detailed feature downsampling without losing important feature information. Unlike methods that focus solely on the maximum value or average of all features, Softpool assigns weights to each feature, giving more importance to features with higher weights, which have a more significant impact on the final output, while features with lower weights have a smaller influence. This approach better preserves the local structure and global features of the point cloud.

First, the input feature dimensions are expanded to 1024, and the Softmax function is used to calculate the weight of each feature element, converting the input feature values into a probability distribution where larger values are assigned higher weights. The formula is as follows:

$$w_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{8}$$

The symbols are defined as follows: x_i is the value of the i -th input feature element, $\sum_{j=1}^{n} e^{x_j}$ represents the sum of the exponential functions of all elements x_j , and w_i is the weight of the element. Based on the calculated weights, the input values are weighted and averaged to produce the pooling result. The formula is:

$$y = \sum_{i=1}^{n} w_i \cdot x_i \tag{9}$$

Finally, the features are fed into FoldingNet, where a series of folding operations progressively refine the point cloud structure layer by layer, generating a complete and detailed point cloud.

D. *Loss Function and Evaluation Metrics*

The loss function for point cloud completion needs to evaluate the difference between the reconstructed point cloud and the original point cloud to ensure the accuracy and quality of the reconstruction. The two most commonly used metric functions are Chamfer Distance (CD) and Earth Mover’s Distance (EMD) [23].

The Chamfer Distance is a computationally efficient method used to measure the difference between the reconstructed point cloud and the original point cloud. Specifically, it calculates the distance from each point in the reconstructed point cloud to its nearest point in the original point cloud, and then averages or sums these minimum distances. Similarly, for each point in the original point cloud, the nearest point in the reconstructed point cloud is found, and the same calculation process is repeated. The sum of these two parts is the Chamfer Distance. On the other hand, the Earth Mover's Distance treats point clouds as probability distributions and measures the minimum "work" required to transform one distribution into another by finding an optimal matching. This method involves solving an optimization problem, which typically has a high computational complexity and is often solved using the Optimal Transport algorithm. Although the Earth Mover's Distance can more accurately reflect the global matching relationship between point clouds, its computational cost is high.

To save computational resources, this paper chooses the Chamfer Distance as the loss function. The Chamfer Distance function not only effectively measures the difference between the reconstructed point cloud and the original point cloud but also improves the effectiveness and performance of point cloud completion while ensuring computational efficiency. The calculation formulas are as follows:

$$J_S = \frac{1}{n_S}\sum_{s\in G} \underset{g\in G}{min} \| s-g \| + \frac{1}{n_G}\sum_{g\in G} \underset{s\in S}{min} \| g-s \| \qquad (10)$$

$$J_k = \frac{1}{n_K}\sum_{k\in K} \underset{g\in G}{min} \| k-g \| + \frac{1}{n_G}\sum_{g\in G} \underset{k\in K}{min} \| g-k \| \qquad (11)$$

Where S represents the set of local center points, containing n_S points, formed by concatenating the predicted local centers and the center points of the input point cloud; K represents the completed point cloud, containing n_K points; and G represents the ground truth point cloud. J_S is used to compare the local centers with the ground truth point cloud, where a high-resolution ground truth point cloud supervises the sparse point cloud, and J_K is used to compare the completed point cloud with the ground truth point cloud. The overall objective is to make the distribution of the predicted point cloud as similar as possible to the ground truth point cloud. Therefore, the final loss function is composed of J_S and J_K:

$$J = J_s + J_k \qquad (12)$$

This experiment uses Chamfer Distance (CD) and F-Score (F1) as evaluation metrics. CD measures global consistency by comparing completed and original point clouds but is less sensitive to local details. In contrast, F-Score evaluates local geometric accuracy and surface matching quality. Together, these metrics assess both global and local completion performance. Lower CD values indicate higher similarity, while higher F-Score values reflect better accuracy and quality.

IV. Experiments and Results Analysis

To comprehensively validate the effectiveness of the proposed model, extensive comparative experiments were conducted on the widely used PCN and ShapeNet55 datasets. Visualized results on the PCN dataset demonstrate the model's ability to restore global structures and fine details, even in challenging scenarios. Additionally, ablation studies were performed to evaluate the contribution of individual components, revealing their importance to the model's performance. The results confirm that MFOSNet excels in handling complex point cloud completion tasks, showcasing its reliability and potential for practical applications.

A. Dataset

1) *PCN Dataset:* This dataset is derived from the ShapeNet database [24] and comprises 8 object categories (e.g., airplanes, cars, chairs, etc.), with a total of over 30,000 point cloud pairs. For each object, the data includes both partial point clouds (2048 points) and complete point clouds (16,384 points). The partial point clouds are generated by simulating common real-world occlusions and noise, while the complete point clouds represent their corresponding lossless versions.

2) *ShapeNet55 Dataset:* As one of the largest publicly available 3D datasets, ShapeNet55 consists of approximately 52,500 3D object samples spanning 55 categories. The dataset is created through a combination of automated 3D scanning and manual CAD modeling, ensuring high diversity and quality. Categories include furniture, vehicles, and various everyday objects, encompassing nearly all types of items encountered in daily life. The extensive quantity and variety of ShapeNet55 make it a valuable resource for research in point cloud completion tasks.

B. Experimental Environment and Parameter Settings

The experimental environment described in this paper uses Ubuntu 20.04 as the operating system, with hardware configured as an i9-13900KF processor, 32GB of memory, and an NVIDIA GeForce RTX 4060TI graphics card. The training environment includes CUDA 11.3, Python 3.8.5, and PyTorch 1.11.0. MFOSNet does not require any pre-training and is an end-to-end trainable model. The AdamW optimizer is employed with an initial learning rate set to 5×10^(-4) and appropriate weight decay. To achieve an optimal balance between performance and resource utilization, the depths of the Transformer encoder and decoder are set to 6 and 8, respectively, with each using 6 attention heads and a hidden dimension of 384. In the KNN operation, the k values are set to 16 and 8.

For the PCN dataset, the batch size is set to 48 with training for 300 epochs, reducing the learning rate by a factor of 0.9 every 21 epochs. For the ShapeNet55 dataset, the batch size is 96 with training for 200 epochs. During training, the loss function is iteratively optimized to update network parameters. At the end of each epoch, the best model is selected using the validation set and evaluated on the test set to assess network performance.

C. Analysis of Experimental Results

1) *Experimental Results of the PCN Dataset:* To comprehensively validate the effectiveness of the MFOSNet model, we conducted comparative experiments on the PCN dataset and analyzed its performance against several state-of-the-art point cloud completion networks. The PCN dataset simulates point cloud deficiencies caused by view occlusions, providing a benchmark for evaluating model performance in reconstructing global structures and handling complex missing regions. This dataset encompasses eight common categories, including airplanes, chairs, and tables, with a standardized point cloud resolution of 2048 points and a moderate data size, ensuring consistent experimental conditions and fair comparisons.

In our experiments, we employed Chamfer Distance (Manhattan norm, CD-L1) as the primary evaluation metric to quantitatively assess the geometric accuracy of the models. Furthermore, we performed visualization experiments to intuitively illustrate the effectiveness of each model in reconstructing global shapes. To comprehensively evaluate model performance, we selected the following representative point cloud completion networks for comparative analysis, including: FoldingNet [3], PCN [1], TopNet [25], GRNet [17], PMP-Net [26], CRN [27] and PoinTr [4].

The experimental results presented in Table I indicate that MFOSNet outperforms PoinTr, CRN, and PMP-Net in terms of average CD values across several categories, such as airplane, table, chair, etc. Specifically, MFOSNet reduces the CD value by 6.5% compared to PoinTr, by 7.8% compared to CRN, and by 10.0% compared to PMP-Net. These results demonstrate that MFOSNet significantly improves the accuracy of point cloud completion tasks, particularly in handling complex shapes.

TABLE I
CHAMFER DISTANCE OF DIFFERENT MODELS UNDER THE PCN (CD-L1)

Categories	FoldingNet	PCN	TopNet	GRNet	PMP-Net	CRN	PoinTr	MFOSNet
plane	9.49	5.50	7.61	6.45	5.65	4.79	4.75	4.54
Cabinet	15.80	22.70	13.31	10.37	11.24	9.97	10.47	9.94
Car	12.61	10.63	10.90	9.45	9.64	8.31	8.68	8.25
Chair	15.55	8.70	13.82	9.41	9.51	9.49	9.39	8.49
Lamp	16.41	11.00	14.44	7.96	6.95	8.94	7.75	6.68
Sofa	15.97	11.34	14.78	10.51	10.83	10.69	10.93	9.96
Table	13.65	11.68	11.22	8.44	8.72	7.81	7.78	7.13
Watercraft	14.99	8.59	11.12	8.04	7.25	8.05	7.29	6.87
Avg	14.31	9.64	12.15	8.83	8.73	8.51	8.38	7.73

Compared to other methods, MFOSNet enhances the capture of point cloud details by extracting and fusing multi-scale features. It also incorporates a Bottleneck Attention Module to further emphasize and refine key detail features, thus optimizing the retention of fine details in the completed point cloud. As a result, MFOSNet significantly improves completion performance, reduces geometric distortions, and further lowers the CD value when processing incomplete point clouds, thereby validating its superior performance in point cloud completion tasks.

The visualization results, as shown in Figure 5, compare the point cloud completion performance of different networks with our proposed MFOSNet method on the PCN dataset. It can be observed that various traditional methods exhibit different characteristics and limitations when handling missing point cloud data:

FoldingNet, PCN, and TopNet employ encoder-decoder architectures for point cloud reconstruction. However, these architectures tend to lose local structural features during the completion process, particularly when dealing with complex geometries or fine details (e.g., the legs of a chair or the wings of an airplane). The completion results from these networks typically show smooth transitions but lack necessary local details, leading to deficiencies in the finer parts of the completed point cloud.

GRNet, based on graph convolutional networks (GCNs), is good at capturing local geometric features. However, it occasionally generates spurious structures that do not align with the original object's geometry, resulting in unnatural geometric distortions or unnecessary details (such as edges or protrusions that should not exist). This can affect the overall quality of the completion.

PMP-Net, while generating denser point clouds to compensate for local missing areas, produces results that appear more refined in some regions (e.g., the surface of a table or an object's surface). However, the overall point cloud distribution is not uniform. In some cases, the completed point cloud exhibits overly dense or concentrated regions, causing an imbalance in the overall shape.

CRN is based on Generative Adversarial Networks (GANs) and aims to improve point cloud completion through adversarial training. While CRN is capable of producing more natural global structures in the completed point clouds, it often falls short in detail recovery. Especially when handling complex geometries, CRN occasionally generates artifacts that do not align with the original object's geometry, manifesting as unnatural geometric distortions or excessive smoothing, such as smooth surfaces that should not exist or unreasonable edges. This phenomenon can lead to a lack of essential local details in the completed point clouds, causing the final result to lose precision and affecting the overall quality and reliability of the completion.

PoinTr, a method based on the Transformer architecture, shows rougher completion results when a significant amount of point cloud data is missing. While it excels in global feature capture, it fails to recover fine details, especially when the input point cloud is sparse. The generated completed point cloud often lacks subtle features, and edges and details appear blurry, failing to effectively restore complex structures.

From the visualization results in Figure 5, it is evident that MFOSNet outperforms these methods. The model not only restores missing local details effectively but also maintains the coherence of the global structure. Specifically, in the completion of complex shapes (e.g., airplane wings and chair legs), the point clouds generated by MFOSNet preserve important geometric features while maintaining a uniform distribution and detailed structure, avoiding excessive smoothing or the introduction of spurious structures. Compared to other methods, MFOSNet's completed point clouds appear more natural, with more accurate detail recovery and a more balanced overall structure. MFOSNet is capable of more accurately restoring details such as object edges, depressions, and protrusions during the completion process. When faced with large areas of missing data, MFOSNet can estimate the global structure of the missing region effectively and fill in the missing points in a reasonable manner, avoiding significant geometric distortions or unnatural transitions. These advantages stem from the innovative design of the MFOSNet model, which captures both local and global features, enabling it to simultaneously optimize completion results at multiple levels.

Overall, MFOSNet not only outperforms traditional methods in accuracy and detail recovery, but also provides more reliable and uniform completion results for point cloud data with large-scale missing regions or complex structures. By effectively capturing both local and global features, the point clouds generated by MFOSNet exhibit higher consistency and balance in their structure. These advantages make MFOSNet highly promising for practical applications such as autonomous driving and robotic grasping, especially in tasks that require high-precision completion, where it performs exceptionally well.

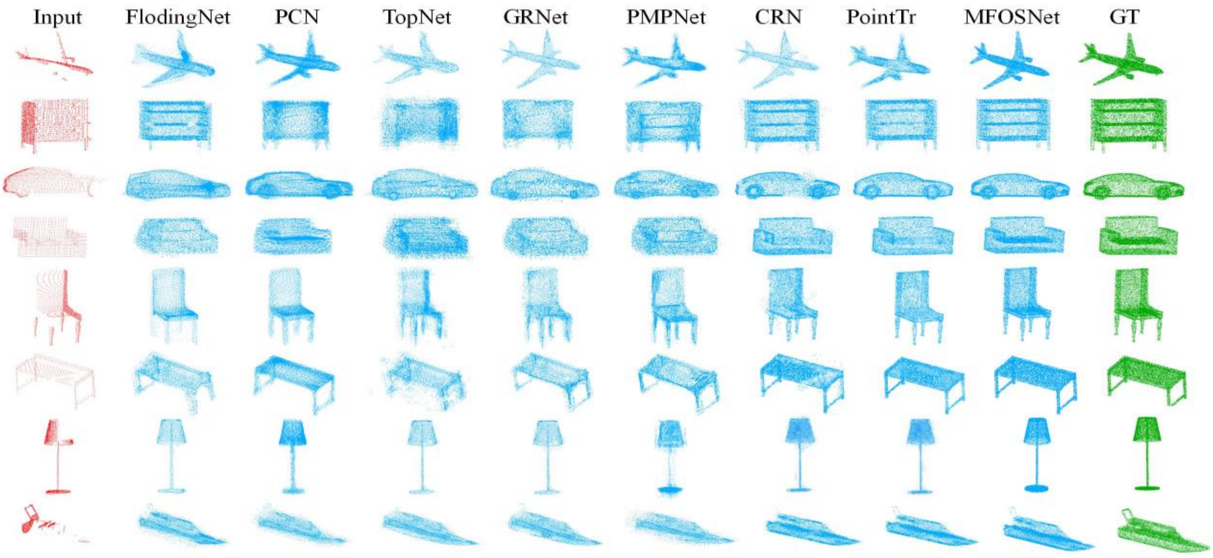

Fig. 5. Visualization results of the PCN dataset

2) *Experimental Results on the ShapeNet55 Dataset:* To evaluate performance on both well-represented and scarce categories, 10 categories are selected from the ShapeNet-55 dataset for experimentation. Among these, Table, Chair, Airplane, Car, and Sofa are classified as well-represented categories. These categories have a large amount of data, providing sufficient training samples for the model to better learn their features. In contrast, Birdhouse, Bag, Remote, Rocket, and Keyboard are classified as scarce categories. Due to the limited data, the model may not fully learn the features of these categories during training, which can impact performance. The completion task of the ShapeNet55 dataset presents significant geometric complexity and category diversity, imposing higher requirements on models' abilities to restore global shapes and capture local details. To evaluate model performance, six representative models—FoldingNet, PCN, TopNet, PFNet, GRNet, and PoinTr—were selected for comparative analysis. These models encompass diverse network architectures and completion strategies, making them highly valuable for comparison.

The experiment categorized point cloud deficiencies into three difficulty levels: simple (25% missing), medium (50% missing), and difficult (75% missing). This classification provides an intuitive assessment of the models' completion capabilities across different scenarios. Additionally, Chamfer Distance (Euclidean norm, CD-L2) and F-Score were adopted as evaluation metrics to measure the geometric distance between the completed and ground-truth point clouds, as well as the accuracy and completeness of the point cloud distribution. The combination of these two metrics enables a comprehensive evaluation of model performance, from global consistency to local detail restoration.

To further analyze the performance of models on categories with abundant and scarce samples, 10 categories were selected from the ShapeNet55 dataset. Table, Chair, Airplane, Car, and Sofa were classified as sample-rich categories, with large amounts of data facilitating models to fully learn their features. In contrast, Birdhouse, Bag, Remote, Rocket, and Keyboard were categorized as sample-scarce categories, where the limited data may constrain the models' completion performance.

Table II highlights the comparative performance of various methods under different difficulty levels, demonstrating that the proposed MFOSNet model consistently outperforms its counterparts. Notably, MFOSNet achieves an impressive score of 0.53 under the easy setting, 0.81 in the medium setting, and a CD value of 1.73 in the difficult setting. These results reveal a decline in performance as difficulty increases; however, the model consistently maintains a high level of accuracy. Furthermore, MFOSNet excels in terms of the F-score, reaching 0.49, showcasing its robust precision and recall capabilities.

Table III provides a detailed breakdown of average results across 10 categories under varying difficulty levels. The data underscores MFOSNet's superior ability to handle point cloud data across diverse perspectives, categories, missing patterns, and degrees of missingness. These findings confirm the model's adaptability and exceptional performance in handling complex and varied environments.

TABLE II
COMPARISON RESULTS OF SHAPENET55 DATASET UNDER DIFFERENT DIFFICULTY LEVELS (CD-L2)

	FoldingNet	PCN	TopNet	PFNet	GRNet	PoinTr	MFOSNet
CD-S	2.67	1.94	2.26	3.83	1.35	0.58	0.53
CD-M	2.66	1.96	2.16	3.87	1.71	0.89	0.81
CD-H	4.05	4.08	4.30	7.97	2.85	1.79	1.73
CD-Avg	3.12	3.12	2.66	2.91	1.97	1.09	1.02
F1	0.08	0.13	0.13	0.34	0.24	0.46	0.49

TABLE III
COMPARISON OF THE MEAN VALUES OF 10 CATEGORIES OF SHAPENET55 DATASET UNDER DIFFERENT LEVELS OF DIFFICULTY (CD-L2)

Categories	FoldingNet	PCN	TopNet	PFNet	GRNet	PoinTr	MFOSNet
Table	2.531	2.130	2.216	3.956	1.632	0.811	0.772
Chair	2.812	2.292	2.530	4.242	1.882	0.953	0.930
Airplane	1.432	1.023	1.144	1.817	1.026	0.446	0.421
Car	1.983	1.850	2.184	2.530	1.641	0.917	0.864
Sofa	2.481	2.062	2.366	3.341	1.727	0.792	0.762
Birdhouse	4.714	4.507	4.830	6.212	2.976	1.868	1.782
Bag	2.792	2.861	2.932	4.960	2.060	0.930	0.880
Remote	1.441	1.333	1.496	2.911	1.098	0.530	0.485
Rocket	1.486	1.324	1.320	2.363	1.036	0.571	0.547
Keyboard	1.242	0.891	0.954	1.295	0.890	0.381	0.342

3) *Ablation Study:* To further evaluate the impact of the dilated convolution layer, bottleneck attention layer, and Softpool module on the experimental results, four sets of ablation experiments were conducted on the PCN dataset. These experiments were designed to assess the effect of removing each of these components on the model's performance and to compare the outcomes with the complete proposed model. By systematically excluding each module, the individual contributions to the overall performance enhancement are clearly identified, providing deeper insights into their significance for point cloud reconstruction.

Table IV details the results of the ablation experiments, showcasing the influence of each module on the model's performance. The findings offer valuable theoretical guidance for further optimization of the model. These experiments not only confirm the critical role of each module but also highlight the effectiveness and robustness of the proposed model in processing point cloud data.

TABLE IV
COMPARISON OF RESULTS OF ABLATION EXPERIMENTS (CD-L1)

Method	Description	CD
(A)	complete model	7.73
(B)	without dilated convolution	7.98
(C)	without bottleneck attention	8.11
(D)	without softpool	8.26

(B) and (C) illustrate the results of removing the dilated convolution and bottleneck attention modules from the multi-scale feature optimization module, respectively. The experimental findings reveal that the average CD values increased by 2.5% and 3.8%, respectively, highlighting the critical role these components play in maintaining low error rates and enhancing the accuracy of reconstruction. (D) demonstrates the effects of replacing Softpool with max pooling, where the average CD value increased significantly by 5.3%. This result underscores the pivotal role of the Softpool module in refining feature representations, preserving geometric fidelity, and ensuring smooth and accurate shape predictions. In contrast, (A) represents the results of the complete MFOSNet model, which achieved the lowest values across all evaluation metrics. This outcome strongly validates the effectiveness of integrating all proposed modules into a cohesive framework, showcasing its superior performance in point cloud completion tasks.

The analysis underscores the substantial contributions of both the multi-scale feature optimization module and the Softpool module to the accuracy of point cloud completion. Together, these three modules significantly bolster the network's capacity to reconstruct detailed and intricate structures, further demonstrating the robustness, efficiency, and adaptability of the proposed model.

Figure 6 provides a visual representation of the ablation experiment results, offering further insight into the roles and significance of the individual modules. In method (D), the absence of the Softpool module results in less smooth predicted shapes, particularly evident in the finer geometries of the airplane and lamp. This observation highlights the critical role of Softpool in extracting and optimizing fine details, which enables the model to produce smoother, more coherent, and visually natural shapes. In method (C), removing the bottleneck attention mechanism leads to blurred and poorly defined boundaries, as seen in objects like the chair and lamp. This limitation further underscores the importance of the bottleneck attention mechanism in emphasizing local features and ensuring the preservation of critical fine details. Method (B), which employs single-scale feature extraction, struggles to capture intricate details effectively and suffers from feature loss, particularly in complex scenes. This limitation is most apparent in challenging scenarios, where the inability to capture multi-scale information leads to suboptimal completion results.

In contrast, method (A), representing the complete MFOSNet model, demonstrates outstanding performance across all metrics. It excels in detail preservation, boundary clarity, and shape smoothness, delivering the most accurate, visually consistent, and appealing reconstructions. The complete model effectively integrates the unique strengths of all modules, optimizing both feature extraction and generation processes to achieve the best possible completion results.

In conclusion, the differences in performance among the various methods in handling shape and detail features vividly highlight the indispensable roles of the individual modules and their synergistic integration. These findings not only confirm the necessity and effectiveness of each component in optimizing model capabilities but also provide valuable insights for future research. By leveraging these observations, further refinements and advancements can be made, enhancing the model's performance and expanding its potential applications in real-world scenarios across diverse domains.

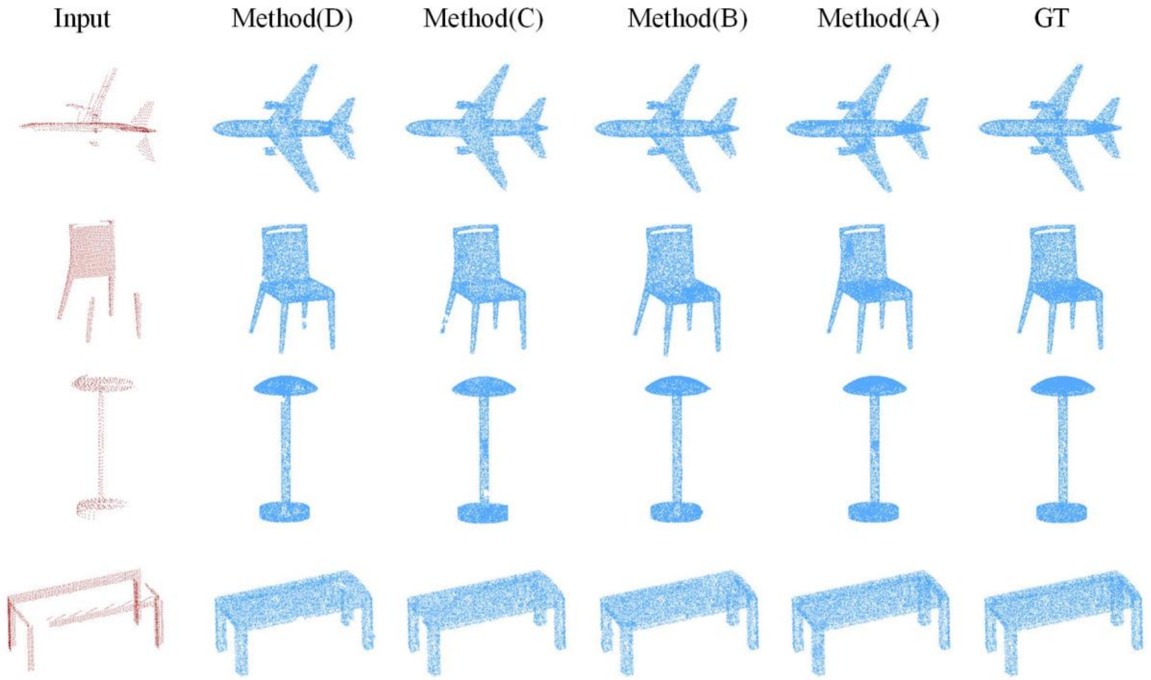

Fig. 6. Visualization results of ablation experiments

V. Conclusion

The proposed MFOSNet introduces a multi-scale feature optimization module that enhances the DGCNN module by integrating dilated convolutions and bottleneck attention mechanisms. This improvement effectively boosts the network's ability to extract local features and understand geometric relationships in incomplete point clouds. Additionally, the network uses Softpool technology instead of traditional max pooling methods, which further retains more detailed features and improves the quality of point cloud completion, achieving the desired results.

Through extensive comparative experiments and ablation studies, the proposed point cloud completion network has demonstrated higher efficiency and accuracy in processing large-scale, complex, and incomplete point cloud data. It has potential applications in various industrial and research fields, such as robot navigation [28], environmental perception for autonomous vehicles [29], and 3D reconstruction for cultural heritage preservation [30] . However, despite the excellent performance of the proposed network, some issues remain to be addressed. For instance, networks based on the Transformer architecture typically have a large number of parameters, which may significantly increase training time. In future research, we plan to implement Gated Attention Units [31] to reduce the complexity of the network, thereby improving both the performance and efficiency of the model.

References

[1] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. 2018 Int. Conf. 3D Vision (3DV)*, Verona, Italy, 2018, pp. 728-737.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 652-660.

[3] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 206-215.

[4] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Los Alamitos, CA, USA, 2021, pp. 12498-12507.

[5] Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, "PF-Net: Point fractal network for 3D point cloud completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 7659-7667.

[6] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv:1511.07122, 2015. [Online]. Available: https://arxiv.org/abs/1511.07122

[7] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck Attention Module," arXiv:1807.06514, 2018. [Online]. Available: https://arxiv.org/abs/1807.06514

[8] S. Thrun and B. Wegbreit, "Shape from symmetry," *in Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Beijing, China, vol. 2, pp. 1824-1831, Oct. 2005.

[9] R. Schnabel, P. Degener, and R. Klein, "Completion and reconstruction with primitive shapes," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 503-512, Mar. 2009.

[10] G. Li, L. Liu, H. Zheng, and N. J. Mitra, "Analysis, reconstruction and manipulation using arterial snakes," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 1866178-1–1866178-10, Dec. 2010.

[11] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," *Comput. Graph.*, vol. 33, no. 0, pp. 187-194, Jul. 1999.

[12] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, and L. J. Guibas, "Example-based 3D scan completion," in *Proc. Symp. Geometry Process.*, Vienna, Austria, 2005, pp. 23-32.

[13] Kang-Xue Yin, Hui Huang, Hao Zhang, Ming-Lun Gong, D. Cohen-Or, and Bao-Quan Chen, "Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 202-1, Nov. 2014.

[14] M. Sung, V. G. Kim, R. Angst, and L. Guibas, "Data-driven structural priors for shape completion," *ACM Trans. Graph. (TOG)*, vol. 34, no. 6, pp. 1-11, Nov. 2015.

[15] Wen-Yue Wang, Qian-Gui Huang, Su-Ya You, Chao Yang, U. Neumann, "Shape inpainting using 3D generative adversarial network and recurrent convolutional networks," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Venice, Italy, Oct. 22-29, 2017, New York: IEEE, 2017, pp. 2298-2306.

[16] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Machine Learning (ICML 2018)*, Stockholm, Sweden, Jul. 10-15, 2018, Red Hook: Curran Associates, Inc., 2018, pp. 67-85.

[17] Hao-Zhe Xie, Hong-Xun Yao, Shang-Chen Zhou, Jia-Geng Mao, Sheng-Ping Zhang, and Wen-Xiu Sun, "GRNet: Gridding residual network for dense point cloud completion," in *Computer Vision – ECCV 2020: Part IX*, Springer, 2020, pp. 365-381.

[18] Cun-Lin Xie, Chu-Xin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen, “Style-based point generator with adversarial rendering for point cloud completion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos: IEEE Computer Society Press, 2021, pp. 4617-4626.

[19] Jing-Bin MA, Dan-Chen Zhu, Ya Zhang, and Xiao-Ming Wang, “Multi-scale feature fusion and contrastive pooling for point cloud completion network,” *Appl. Res. Comput.*, vol. 41, no. 2, pp. 635-640, Jun. 2024.

[20] Yue Wang, Yong-Bin Sun, Zi-Wei Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic Graph CNN for learning on point clouds,” *ACM Trans. Graph. (TOG)*, vol. 38, no. 5, pp. 146, Jun. 2019.

[21] C. R. Qi, Li Yi, Hao Su, and L. J. Guibas, “PointNet++: deep hierarchical feature learning on point sets in a metric space,” in *Proc. 31st Int. Conf. Neural Information Processing Systems (NeurIPS)*, Red Hook: Curran Associates Inc., 2017, pp. 5105-5114.

[22] T. Abeywickrama, M. A. Cheema, and D. Taniar, “k-Nearest Neighbors on Road Networks: A Journey in Experimentation and In-Memory Implementation,” *Proc. VLDB Endowment*, vol. 9, no. 6, pp. 492-503, Jan. 2016.

[23] Hao-Qiang Fan, Hao Su, and L. Guibas, “A Point Set Generation Network for 3D Object Reconstruction from a Single Image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 21-26, 2017, New York: IEEE, 2017, pp. 2463-2471.

[24] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Qi-Xing. Huang, Zi-Mo Li, et al., “ShapeNet: An information-rich 3D model repository,” *arXiv preprint* arXiv, May. 18, 2014. (Online). Accessed: Dec. 9, 2015. Available: http://arxiv.org/abs/1512.03012.

[25] L. P. Tchapmi, V. Kosaraju, H. Rezatofighi, I. Reid, and S. Savarese, “TopNet: Structural point cloud decoder,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 383-392.

[26] Xin Wen, Peng Xiang, Zhi-Zhong Han, Yan-Pei Cao, Peng-Fei Wan, Wen Zheng, Yu-Shen Liu, "PMP-Net: Point cloud completion by learning multi-step point moving paths," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, USA, June. 20-25, 2021, pp. 214-242.

[27] Xiao-Gang, M. H. Ang, and G. H. Lee, “Cascaded Refinement Network for Point Cloud Completion,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 14-19, 2020, New York: IEEE, 2020, pp. 790-799.

[28] Dong Liu, Fei Chen, Qiang Zou, and Ming Cong, “Mobile robot navigation based on situational experience and sparse point cloud,” *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 48, no. 9, pp. 25-30, Sep. 2020.

[29] Jian-Hong Ma, Xi-Yao Wang, Yong-Xia Chen, and Nan Lin, “A Review of Research on Image and Point Cloud Fusion Methods in Automatic Driving,” *Journal of Zhengzhou University (Natural Science Edition)*, vol. 54, no. 6, pp. 24-33, Dec. 2022.

[30] Bin Li, Cheng-Qi Cheng, and Qi-Aan Duan, “Application and Discussion on Point Cloud Technology in Architectural Culture Heritage Protection,” *Urban Surveying and Mapping*, 2014, no. 2, pp. 99-102.

[31] Wei-Zhe Hua, Zi-Hang Dai, Han-Xiao Liu, and Quoc V. Le, “Transformer Quality in Linear Time,” in *Proc. Int. Conf. Mach. Learn. (ICML 2022)*, Baltimore, MD, USA, Jul. 17-23, 2022, Part 11 of 33, pp. 9099-9117.