IE-PMMA: Point Cloud Completion Through Inverse Edge-aware Upsampling and Precise Multi-Modal Feature Alignment

Ran Jia[1] , **Junpeng Xue**[1,*] , **Shuai Ma**[2] , **Wenbo Lu**[1] and **Kelei Wang**[1]
[1]Key Laboratory of Advanced Spatial Mechanism and Intelligent Spacecraft, Ministry of Education,
School of Aeronautics and Astronautics, Sichuan University
[2]ChengDu Aircraft Industrial (Group) Co., Ltd.
rjia@stu.scu.edu.cn, jpxue@scu.edu.cn

Abstract

Point cloud completion is a crucial task in 3D computer vision. Multi-modal completion approaches have gained attention among the popular two-stage point cloud completion methods. However, there is a notable lack of research focused on accurately aligning data from different modalities within these methods. Additionally, in other point cloud-based tasks, edge point information often provides unexpected positive contributions. In this paper, we propose a novel point cloud completion method that leverages edge point information for the first time in the completion task, which also addresses the precise alignment of multi-modal data. In particular, we implement a two-step local-to-global module to achieve better alignment of multi-modal data during the preliminary point cloud generation process. Besides, we introduce a new spatial representation structure capable of extracting a fixed number of edge points. Moreover, with the assistance of edge information, we further design an inverse edge-aware upsampler to refine the point cloud. We evaluate our method on three typical datasets, and the results demonstrate that our IE-PMMA outperforms the existing state-of-the-art methods quantitatively and visually.

1 Introduction

The development of scanner technology has led to the widespread use of 3D point clouds in various applications. However, the point clouds from scanners are often sparse and incomplete[Geiger *et al.*, 2013] due to limited perspectives[Jia *et al.*, 2024] and scanner resolution. This makes it challenging to use them directly in applications like reconstruction of archaeological[Jaklič *et al.*, 2015], autonomous vehicle detection[Ma *et al.*, 2022], virtual reality scan modeling[Kim *et al.*, 2019], and other point-based downstream tasks[Tesema *et al.*, 2023]. As a result, completing point clouds has become an increasingly important task in 3D computer vision.

Currently, mainstream point cloud completion methods are typically based on a two-stage framework. In the first stage, a coarse point cloud with a complete shape (referred to as seed points) is generated from the input partial points, known as the seed points generation step. In the second stage, based on the extracted point cloud features, the seed points are upsampled to the target resolution in order to generate a complete and dense point cloud. This stage is referred to as the seed points upsampling step.

With the widespread application of multi-modal ideas, 2D images have become common as supplementary data in point cloud completion tasks. ViPC[Zhang *et al.*, 2021] is the pioneer in the use of 2D data for completion tasks, taking the missing crucial global structure information from an extra single-view image. CSDN[Zhu *et al.*, 2023c] estimates the global and local geometric information of point clouds from 2D images through a dual-refinement network. Nevertheless, these methods usually use projection relations or attention mechanisms to fuse multimodal data. Such fusion methods can be limited and imprecise, not fully taking advantage of the correlation between 2D and 3D data.

Therefore, inspired by the feature fusion method in BEV[Chen *et al.*, 2022] tasks, we propose a two-stage precise multi-modal data alignment method, which is divided into two steps: local feature alignment and global feature fusion. Firstly, the projection depth maps under different viewpoints are obtained through the partial points, and the 2D image features are extracted. At the local scale, the image features corresponding to each point in the point cloud are obtained through projection relations and bilinear interpolation. Then, the scale factor adjusts the feature's magnitude to get accurate 2D features. The 3D features are extracted directly via PointNet++[Qi *et al.*, 2017b]. At the global scale, we continue to use the attention mechanism-based fusion method in previous work to further fuse features.

In addition, we notice that in many works related to point cloud, edge information has been introduced as a way to augment metadata, such as point cloud generation[Liu *et al.*, 2022], classification[Wu *et al.*, 2023], and other tasks. Many works suggest that the edge points have rich geometric details, which can provide more information about the local structure of the point cloud. Therefore, we hope to introduce the idea of edge-aware to the point cloud completion task.

To achieve edge-aware completion, we first design a new spatial representation method for extracting edge points called Edge Probability Volume (EPV). This method divides the space containing the point cloud into grids. Each voxel contains two values: the coordinates of the point within that voxel and the probability of the voxel being an edge voxel.

We applied the edge points extracted by EPV during the seed points upsampling process. Calculate the chamfer distance between edge points and seed points, as well as between input points and seed points, to represent edge importance and missing importance, respectively. The self-attention module is employed to achieve edge-aware upsampling. However, in practice, we discover that inverse edge-aware upsampling is more effective. Specifically, the closer a point is to the edge points, the less important it is as a seed point. By utilizing the cross-attention operation between the extracted aware features and the input partial point features, we obtain the displacement of point cloud coordinates through two decoders and MLP layers.

We experiment with our method on three common datasets: PCN, ShapeNet34/55, and KITTI. The results show that our method achieves SOTA in metrics of chamfer distance, density-aware chamfer distance, and F1 score. Our main contributions can be summarized as:

1. We propose a local-to-global feature extraction module based on precise multi-modal data alignment. This module fully integrates 2D and 3D features and effectively leverages the information contained in the depth map.
2. We introduce an edge point extraction method that can extract a fixed number of edge points, effectively reducing the impact of point cloud noise and adapting to the point cloud completion task.
3. We apply edge point information to the point cloud completion task for the first time. We utilize edge information during the upsampling process of seed points to achieve inverse edge-aware upsampling.

2 Related Work

2.1 Two-Stage Point Cloud Completion

The existing point cloud completion methods based on deep learning are generally two-stage completion frameworks. Firstly, a complete seed point cloud is generated from the input partial point cloud, and then the final completion result is obtained through the upsampling process. PCN[Yuan *et al.*, 2018] is an outstanding contribution to the two-stage point cloud completion framework, which uses the PointNet[Qi *et al.*, 2017a] layer to extract global features and generate seed points with MLPs, then uses Folding-based[Yang *et al.*, 2018] upsampling block to upsample seed points. PoinTr[Yu *et al.*, 2021] uses a lightweight DGCNN[Wang *et al.*, 2019] to extract point cloud features, a geometry-aware Transformer to predict missing point proxies, and upsamples seed points using Folding block as in PCN. Snowflakenet[Xiang *et al.*, 2021] uses PointNet++[Qi *et al.*, 2017b] to extract features and generate seed points with point-wise splitting operations and MLPs. SeedFormer[Zhou *et al.*, 2022], AnchorFormer[Chen *et al.*, 2023], and ProxyFormer[Li *et al.*, 2023] design unique point cloud representations to improve the ability to extract local geometric information around seed points. ODGNet[Cai *et al.*, 2024] designed a shape prior dictionary-guided seed point generation U-Net, and used the Upsampling Transformer in SeedFormer for seed points upsampling. SVDFormer[Zhu *et al.*, 2023b] applies the depth projection of the partial points to the point cloud completion and uses the two-dimensional depth map to complete. On this basis, GeoFormer[Yu *et al.*, 2024] applies the rich features contained in the three-channel canonical coordinate maps to improve the multi-modal point cloud completion. T-CorresNet[Duan *et al.*, 2025] generates point proxies with the help of the Gaussian spherical template and upsamples them with the Folding block.

These methods can effectively complete the missing parts of the point cloud, but the current completion works have relatively few studies that incorporate multi-modal data, especially those that focus on further aligning data from different modalities. Furthermore, current research lacks exploration of the usage of the edge points.

2.2 Point Cloud Edge Extraction

The method of point cloud edge extraction can be divided into geometry-based methods and learning-based methods. Geometry-based methods are mainly based on local geometric features, such as the eigenstructure of the covariance matrix[Gumhold *et al.*, 2001], normals[Weber *et al.*, 2010], curvatures[Hackel *et al.*, 2016], or other statistical metrics[Zhu *et al.*, 2023a]. There are also some point cloud edge feature extraction methods based on graph Fourier transform[Chen *et al.*, 2017; Deng *et al.*, 2022]. These geometry-based methods are able to extract approximate edge points, but they are less accurate.

With the development of deep learning, learning-based point cloud edge extraction methods have emerged. The earliest methods are based on point cloud classification. PIE-Net[Wang *et al.*, 2020], EDC-Net[Bazazian and Parés, 2021], and PCEDNet[Himeur *et al.*, 2021] classify point clouds into edge points, non-edge points, and corner points based on the sharpness labels of points in the dataset to extract edge points and parametric lines. EC-Net[Yu *et al.*, 2018] and DEF[Matveev *et al.*, 2022] regress the distance between the sampled points and the feature line to extract the identified edge points. These methods are explicit methods that are easily affected by point cloud noise[Zhu *et al.*, 2023a]. NerVE[Zhu *et al.*, 2023a] divides the point cloud space into grids and binary classifies whether each voxel is an edge voxel or not. NEF[Ye *et al.*, 2023] used the image of the object model to extract edge information and designed an edge neural radiance field. PBWR[Huang *et al.*, 2024] extracts the edge of LiDAR point clouds through a transformer to obtain edge regression results.

These methods can extract edge points better than geometry-based methods, but the number of edge points is not fixed, making them unsuitable as a data augmentation scheme for downstream point cloud tasks.

3 Method

3.1 Overview

We follow the commonly used two-stage point cloud completion paradigm, first generating the seed points with a complete shape from the input partial points and then upsampling

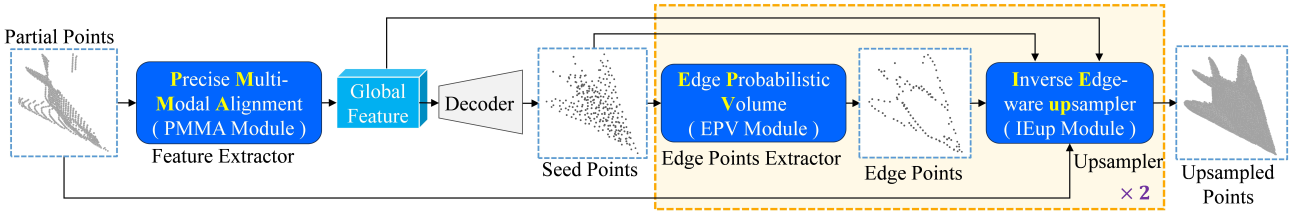
Partial Points
Precise Multi-Modal Alignment (PMMA Module)
Feature Extractor
Global Feature
Decoder
Seed Points
Edge Probabilistic Volume (EPV Module)
Edge Points Extractor
Edge Points
Inverse Edge-ware upsampler (IEup Module)
Upsampler
× 2
Upsampled Points

Figure 1: The pipeline of our method. Firstly, the input partial points are fed into the Precise Multi-Modal Alignment Module to extract the global feature, while the complete seed points are obtained by a decoder. The edge points of seed points are extracted through the Edge Probability Volume. Then, they are inputted into the Inverse Edge-aware upsampler, together with the seed points and global feature. The dense and complete point cloud is finally obtained by a two-step upsampling process.

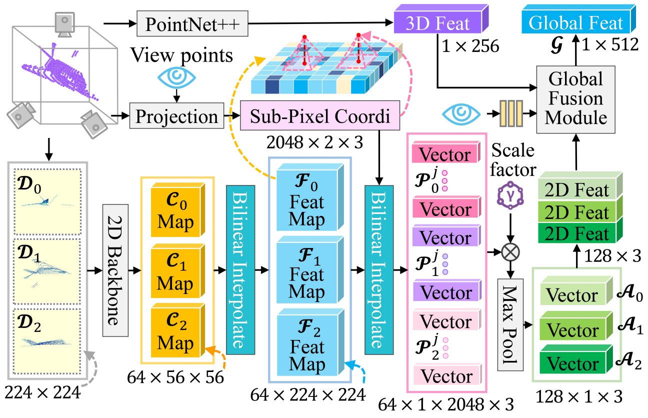

Figure 2: The architecture of the local PMMA module. At the local scale, the PMMA module first realizes the point-to-pixel local feature alignment through camera projection relationship and bilinear interpolation, then shrinks the features through the scale factor.

the seed points to the target resolution, as shown in Figure 1. Specifically, the global feature is extracted from input partial points through the **P**recise **M**ulti-**M**odal **A**lignment feature extractor (called PMMA module). The 3D coordinates of the seed points are regressed by a group of MLP layers. In order to take advantage of the information on the edge points, we first obtain the edge seed points through the pre-trained **E**dge **P**robability **V**olume (called EPV module). The partial points, seed points, edge seed points, and global feature are taken as input of the **I**nverse **E**dge-aware upsampler (called IEup module), for the upsampling of seed points. In line with other coarse-to-fine upsampling methods, our IEup module is also performed twice to achieve the coarse-to-fine upsampling of seed points.

3.2 Seed Points Generation With Precise Multi-Modal Feature Alignment

The first stage of the two-stage completion framework is the generation of seed points with a completed shape. Currently, multi-modal data is used in many point cloud completion works. Among them, the works represented by [Zhu *et al.*, 2023b] project the partial point cloud, and the obtained depth maps are used as the image modal data. The alignment between the 3D point cloud features and the 2D image features is then achieved through the cross-attention mechanism. This approach only aligns features on a global scale and can't achieve the local accurate point-to-pixel feature alignment. Inspired by AutoAlignv2[Chen *et al.*, 2022], we design a local-to-global feature fusion module, which enables precise multi-modal feature alignment.

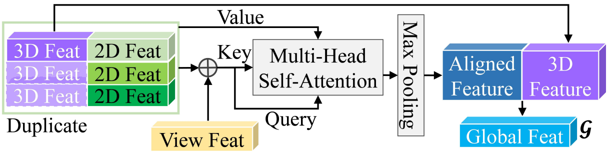

Figure 3: The architecture of the global PMMA module. The attention mechanism achieves feature alignment at the global scale.

At the local scale, as show in Figure 2, firstly, the input partial point cloud is projected according to the coordinates of viewpoints as the camera position to obtain the depth map $\mathcal{D}_i$ in three directions, i indicates the number of viewpoints. Afterward, use 2D Backbone to extract features from the depth map and obtain a low-resolution coarse feature map $\mathcal{C}_i$. The 2D Backbone here is a simplified ResNet18[He *et al.*, 2016]. Then, the low-resolution basic feature map is upsampled to the fine feature map $\mathcal{F}_i$ through bilinear interpolation, which is consistent with the resolution of the depth map $\mathcal{D}_i$. At the same time, the point cloud is projected according to three viewpoints to calculate the sub-pixel coordinates corresponding to each point. The image feature $\mathcal{P}_i^j$ corresponding to each point is extracted on the fine feature map through bilinear interpolation, while j represents the index of the point.

The extracted features are scaled through scale factor γ, and the processed features are fed into the max pooling layer for aggregation and obtaining the image features $\mathcal{A}_i$ that are accurately aligned with the coordinates of the 3D points. The formula of this process can be expressed as:

$$\mathcal{A}_i = \text{MaxPool}(\gamma \sum_{j=1}^{\mathcal{N}} \mathcal{P}_i^j), \text{where } i = 0, 1, 2 \text{ and } \gamma = \frac{1}{\mathcal{N}} \quad (1)$$

where $\mathcal{N}$ represents the number of points. Then cascading $\mathcal{A}_i$ from all viewpoints to get the final 2D feature. The corresponding 3D features are directly extracted through PointNet++[Qi *et al.*, 2017b]. Finally, these two different dimensional features and the viewpoints are inputted into the

Global Fusion Module to get the global feature $\mathcal{G}$.

At the global scale, as shown in Figure 3, we use the attention mechanism[Zhu *et al.*, 2023b] to achieve the re-fusion of multi-view 2D image features and 3D point cloud features. First, the 3D features are duplicated and concatenated with the 2D features. The view features are added to the 2D & 3D concatenation features to serve as the Key and Query of the multi-head self-attention, while the concatenation features as the Value. The alignment features obtained by the max pooling layer are then concatenated with the 3D features to obtain the global feature $\mathcal{G}$, which is fed into MLP layers to regress the coordinates of seed points.

3.3 Edge Points Extraction Based on Edge Probability Volume

In the fields of point cloud generation [Liu *et al.*, 2022], classification, retrieval [Wu *et al.*, 2023; Wen *et al.*, 2023], and reconstruction [Wen *et al.*, 2023], the edge information of the point cloud can enhance the metadata. This is due to the fact that the edge points contain a wealth of sharp details. It can be inferred that the edge information of the point cloud has a positive effect on a variety of point cloud tasks. Therefore, we hope to use the edge information of the point cloud as a new exploration in the point cloud completion task.

To adapt to the point cloud completion task, we propose a new representation called Edge Probability Volume (EPV) to predict and rank the probability of whether one point is an edge point. This section mainly introduces the definition of EPV and the method of extracting edge points.

To solve the problem that the classification-based point cloud edge extraction methods [Wang *et al.*, 2020; Bazazian and Parés, 2021; Himeur *et al.*, 2021] are susceptible to noise, NerVE [Zhu *et al.*, 2023a] follows the voxel representation of a 3D shape [Choy *et al.*, 2016] to discretize continuous 3D curves into a volumetric grid. On this basis, we improve the binary classification of edge points to probabilistic regression, which realizes the orderliness of point cloud probability so that a fixed number of edge points can be obtained.

As shown in Figure 4, a simplified PointNet++[Qi *et al.*, 2017b] module and a 3D CNN module are used to obtain the feature grid of the point cloud, which has the same resolution as EPV output. After that, two individual decoders are used to predict the attributes of each voxel. In the first step, one MLP regresses the probability $p_{i,j,k}$ of whether a voxel is an edge voxel. In the second step, the other MLP calculates the coordinates $\mathbf{x}_{i,j,k}$ of the point in each voxel. The process can be expressed as:

$$p_{i,j,k} = MLP_{prob}(\boldsymbol{\mathcal{E}}_{i,j,k}) \tag{2}$$

$$\mathbf{x}_{i,j,k} = MLP_{pos}(\boldsymbol{\mathcal{E}}_{i,j,k}), \text{where } p_{i,j,k} > \xi \tag{3}$$

where $\boldsymbol{\mathcal{E}}_{i,j,k}$ is the voxel feature and ξ is the threshold. Only voxels with a probability greater than the threshold will proceed to the coordinate prediction step.

Since the ABC dataset[Koch *et al.*, 2019] we used for training only labels the sharpness of each point, without the probability of whether a point is an edge point. Therefore, we have to define the calculation method of the probability to get the ground truth of the probability.

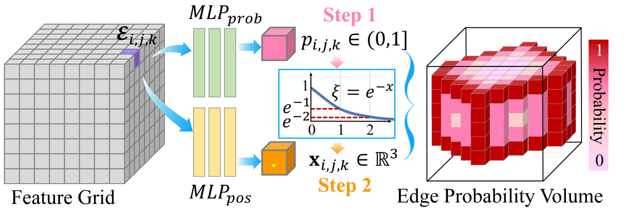

Figure 4: Composition of the spatial representation of the Edge Probability Volume. The EPV module divides the space into grids and extracts the corresponding feature grid. Each voxel is regressed by two different MLPs to obtain the probability that the current voxel is an edge voxel and the coordinates of the edge point in it.

According to the first law of geography, the closer a point cloud is to an edge point, the higher the probability that it is an edge point. As the distance from the edge point increases, the probability of the point cloud being an edge point decreases. Therefore, for each voxel $v_{i,j,k}$, first retrieve the nearest edge voxel $e_{a,b,c}$ to it, then define the distance between them as:

$$d_{i,j,k} = \sqrt{(i-a)^2 + (j-b)^2 + (k-c)^2} \tag{4}$$

In order to convert the distance into a probability in the range of $(0, 1]$, we can use the function $f(x) = e^{-x}$. In this way, the converted distance $f(d_{i,j,k})$ is the ground truth of probability $p^{gt}_{i,j,k}$. In the training process, we chose MSE and $L1$ loss as the loss function to train the probability and coordinates prediction model, respectively , which can be expressed as:

$$\mathcal{L}_{prob} = \frac{1}{N^3} \sum_{i,j,k=0}^{N} \left(p^{gt}_{i,j,k} - p_{i,j,k}\right)^2 \tag{5}$$

$$\mathcal{L}_{pos} = \frac{1}{\|\mathbf{x}_{i,j,k}\|} \sum_{p_{i,j,k} > \xi} \|\mathbf{x}^{gt}_{i,j,k} - \mathbf{x}_{i,j,k}\|_1 \tag{6}$$

where N represents the resolution of the EPV in one direction, $\mathcal{L}_{prob}$ represents the probability prediction loss, and $\mathcal{L}_{pos}$ represents the coordinate prediction loss.

With the pre-trained EPV model, we can predict the probability of whether a voxel is an edge voxel. After sorting the probability values, we can extract the first M voxels with the highest probability value. The points in those voxels are taken as the edge points of the point cloud. M can be modified according to the requirements of the point cloud completion task. In this paper, we choose $32 \times 32 \times 32$ as the resolution to pre-train the EPV model.

3.4 Seed Points Upsampling Based on Inverse Edge-Aware Upsampler

In the previously mentioned point cloud tasks, the premise of introducing point cloud edge information to enhance the point cloud data is the integrity of the point cloud. The edges of the point cloud are meaningful only if the point cloud is complete. In the two-stage point cloud completion method, the input of the seed point generation stage is an incomplete point cloud, and the input of the upsampling stage is the seed points with a relatively complete overall shape. Therefore,

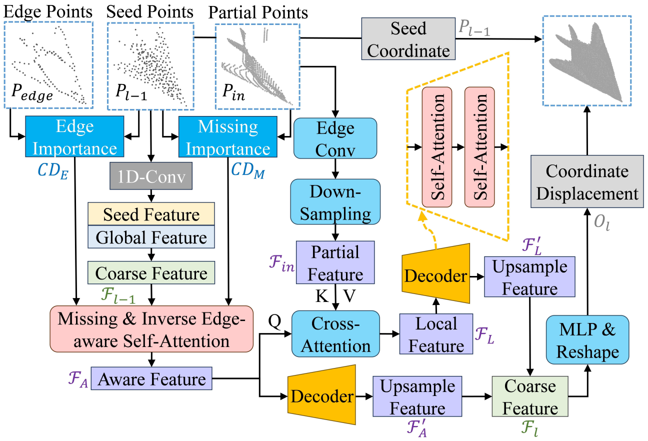

Figure 5: The architecture of Inverse Edge-aware upsampler. The chamfer distances between the seed points and the edge points, as well as the partial points, are calculated to realize the inverse edge-aware feature extraction. The aware features and partial features are fed into two decoders to regress the displacement of seed points.

we chose to introduce point cloud edge information in the upsampling process, as shown in Figure 5.

Specifically, for the generated seed points, we first apply the EPV described in 3.3 to extract the edge points. The first step of upsampling is the Missing & Inverse Edge-aware Self-Attention Module. We define the missing importance as the chamfer distance between the seed points and the partial points, and the greater the chamfer distance from the input point cloud, the greater the missing importance. Similarly, we define the inverse edge importance as the chamfer distance between the seed points and the edge points, and the greater the chamfer distance from the edge points, the greater the inverse edge importance.

It is speculated that point clouds near edge points should be given greater importance based on their impact on other applications. However, our practical findings indicate that using inverse edge importance can actually enhance point cloud completion performance. In other words, the farther a point cloud is from the edge point, the greater importance it is assigned. The missing importance and inverse edge importance can be expressed as:

$$MI\left(p_s\right) = \min_{p_{in}\in P_{in}} \|p_s - p_{in}\|,\ p_s \in P_{seed} \tag{7}$$

$$IEI\left(p_s\right) = \min_{p_e\in P_{edge}} \|p_s - p_e\|,\ p_s \in P_{seed} \tag{8}$$

where the p_s represents the point in the set of seed points P_{seed}, p_{in} represents the point in the set of input points P_{input}, p_e represents the point in the set of edge points P_{edge}. MI represents the Missing Importance of a point, while IEI represents the Inverse Edge Importance. The Aware Importance can be expressed as:

$$AI = Sin\left(\frac{1}{\mu}\left(MI\left(p_s\right) + IEI\left(p_s\right)\right)\right),\ p_s \in P_{seed} \tag{9}$$

where μ is a scaling coefficient set as 0.2, Sin is the sinusoidal function[Vaswani, 2017]. The aware feature is obtained through multi-head attention layers, which can be expressed as:

$$\mathcal{F}_A = MHSA(\mathcal{F}_{l-1}, AI) \tag{10}$$

The aware feature $\mathcal{F}_A$ and partial feature $\mathcal{F}_{in}$, which is extracted from the input point cloud, are fed in the cross-attention layer to obtain the local feature $\mathcal{F}_L$. $\mathcal{F}_A$ and $\mathcal{F}_L$ are fed in two decoders respectively to obtain the upsampled features $\mathcal{F}'_A$ and $\mathcal{F}'_L$. The coarse feature $\mathcal{F}_l$ of the l-th upsampling process is obtained by cascading $\mathcal{F}'_A$ and $\mathcal{F}'_L$. The coarse features are fed into the MLP layers to obtain the displacement of point cloud coordinates at the l-th step. Such a process will be performed twice to achieve the coarse-to-fine upsampling.

3.5 Loss Function

As with other methods, we use the chamfer distance as the loss function. CD loss is defined as follows:

$$CD = \frac{1}{N_1}\sum_{o\in O}\min_{g\in GT}\|o-g\|_2^2 + \frac{1}{N_2}\sum_{g\in GT}\min_{o\in O}\|o-g\|_2^2 \tag{11}$$

where O is the predicted point cloud with N_1 points. GT is the ground truth point cloud with N_2 points. As in the previous work, we also use CD-L1 and CD-L2 as the loss functions for different datasets, respectively, where CD-L2 is the same as CD, and CD-L1 takes the square root of L2-Norm and divides it by 2. The total loss is defined as:

$$\begin{aligned}\mathcal{L} &= CD(P_{seed}, \text{FPS}(P_{gt}, N_s)) \\ &+ CD(P_{l=1}, \text{FPS}(P_{gt}, N_1)) + CD(P_{l=2}, \text{FPS}(P_{gt}, N_2))\end{aligned} \tag{12}$$

The FPS represents the farthest point sampling function, the first parameter represents the point cloud to be sampled, and the second parameter represents the number of sampling points. P_{seed} represents the seed point cloud, P_{gt} represents the ground truth, and $P_{l=1}$ and $P_{l=2}$ represent the corresponding point clouds obtained through two upsampling stages.

4 Experiments

4.1 Dataset and Evaluation Metrics

Dataset. In order to verify the point cloud completion effect of our IE-PMMA in different scenarios, we tested it on common datasets **PCN**, **ShapeNet34/55** and **KITTI** and evaluated them with a number of evaluation metrics. PCN[Yuan *et al.*, 2018] dataset is a subset of the ShapeNet[Chang *et al.*, 2015] dataset, which contains 8 types of objects and 30,974 point clouds. ShapeNet34/55 is proposed by PointTr[Yu *et al.*, 2021]. The ShpeNet55 dataset contains 55 categories of point cloud data, and the ShapeNet34 dataset contains 34 visible category point clouds and 21 invisible category point clouds. We further tested our results on the KITTI[Geiger *et al.*, 2013] dataset. Following the previous method, we only focus on the points in the bounding box of the object marked as a car, with a total of 2483 incomplete point clouds and no ground truth.

Metrics. In order to comprehensively evaluate our method, we used Chamfer Distance (CD), Density Chamfer Distance(DCD), and F1 Score as evaluation metrics on the PCN and ShapeNet34/55 datasets. For the KITTI dataset, we used Fidelity and Minimal Matching Distance (MMD) as the metrics since KITTI is without ground truth.

Methods	CD-8categories↓								CD-Avg↓	DCD-Avg↓	F1↑
	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat			
GRNet[ECCV'20]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83	0.622	0.708
PMP-Net++[TPAMI'23]	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56	0.611	0.781
PoinTr[ICCV'21]	4.75	10.47	8.86	9.39	7.75	10.93	7.78	7.29	8.38	0.611	0.745
SnowFlake[PAMI'23]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21	0.585	0.801
ProxyFormer[CVPR'23]	4.01	9.01	7.88	7.11	5.35	8.77	6.03	5.98	6.77	0.577	-
SeedFormer[ECCV'22]	3.85	9.05	8.06	7.06	**5.21**	8.85	6.05	5.85	6.74	0.583	0.818
SVDFormer[ICCV'23]	**3.62**	8.79	**7.46**	6.91	5.33	8.49	5.90	5.83	6.54	**0.536**	**0.841**
ODGNet[AAAI'24]	3.77	**8.77**	7.56	**6.84**	**5.09**	**8.47**	**5.84**	**5.66**	**6.50**	0.542	0.833
Ours	**3.67**	**8.66**	**7.37**	**6.90**	5.31	**8.32**	**5.90**	**5.72**	**6.48**	**0.532**	**0.844**

Table 1: The quantitative results on the PCN dataset (CD-L1$\times 10^{-3}$).

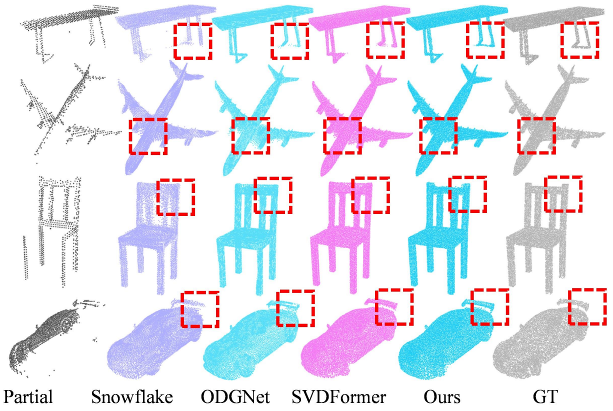

Figure 6: The visualization results on the PCN dataset.

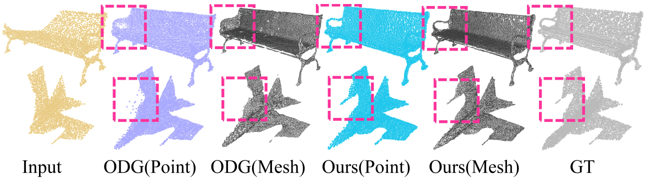

Figure 7: The visualization results on the ShapeNet55 dataset.

Methods	CD-S↓	CD-M↓	CD-H↓	CD-Avg↓	DCD-Avg↓	F1↑
PCN	1.94	1.96	4.08	2.66	0.618	0.133
GRNet	1.35	1.71	2.85	1.97	0.592	0.238
PoinTr	0.58	0.88	1.79	1.09	0.575	0.464
ProxyFormer	0.49	0.75	1.55	0.93	0.549	**0.483**
SeedFormer	0.50	0.77	1.49	0.92	0.558	0.472
SVDFormer	0.48	0.70	**1.30**	0.83	**0.541**	0.451
ODGNet	**0.47**	**0.70**	1.32	**0.83**	0.583	0.437
Ours	**0.42**	**0.60**	**1.07**	**0.70**	**0.527**	**0.480**

Table 2: The quantitative results on the ShapeNet55 dataset.

4.2 Comparison With SOTA Methods

Evaluation on PCN dataset. We conduct experiments on the PCN dataset and compare them with 8 recent point cloud completion methods as shown in Table 1. The results show that IE-PMMA exhibits SOTA results on all metrics. Especially on the DCD, our method is 1×10^{-2} higher than ODGNet. Figure 6 shows the visual comparison results of our method and other methods on the PCN dataset. Our IE-PMMA works best on table legs, aircraft engines, chair backs, and car tails. This is due to the fact that the IEup module makes the model pay more attention to the global structure. At the same time, the PMMA module promotes the generation of the local detailed structure.

Evaluation on ShapeNet34/55 dataset. To investigate the generalization of the method, we test IE-PMMA on ShapeNet34/55. The first is the ShapeNet55 dataset with 55 categories of objects, as shown in Table 2. Our method outperforms the SOTA methods on three representative metrics. As for the ShapeNet34-21 dataset, we surpass the SOTA approach on CD metrics as shown in Table 3. Figure 7 shows the visualization of our method on the Shapenet55 dataset. In addition to visualizing the point cloud results, we also used the ball pivoting algorithm to reconstruct the surface to reflect the representation power of the point cloud. Compared with the SOTA methods, our method is able to accurately recover the local structure of the point cloud, such as the armrests of the chair and the detailed wings of the aircraft. From the results of surface reconstruction, the point clouds completed by our method also have a stronger geometric representation ability.

Evaluation on KITTI dataset. To verify the effectiveness of our method on the real-world scanned dataset, we conduct experiments on the KITTI dataset. We train on the car category of the PCN dataset and recover the full shape from the scanned incomplete vehicle point cloud. Our method surpasses the SOTA methods on the Fidelity metric, as shown in Table 4. Figure 8 shows that our point clouds have more complete shapes and less noise.

4.3 Ablation Experiments

Ablation on IEup Module. In order to verify the role of the Inverse Edge-aware upsampler in the point cloud completion process, we perform ablation experiments corresponding to model A and model B in Table 5. After adding the IEup module, all of the metrics have been significantly improved. In order to show the execution of the IEup module more concretely, we visualize the importance of the point cloud during the upsampling process and the edge points extracted by EPV in Figure 9. The orange points are the edge points, while magenta indicates the greater importance of point clouds, and cyan is the opposite. It can be found that the Inverse Edge-aware upsampler gives more importance to the point cloud far away from the sharp edge of the point cloud, which makes the upsampling process pay more attention to the global shape of the object, rather than over-fitting the local details during the training process. In addition, the EPV spatial representation

Methods	34 seen categories					
	CD-S↓	CD-M↓	CD-H↓	CD-Avg↓	DCD-Avg↓	F1↑
PCN	1.87	1.81	2.97	2.22	0.624	0.150
GRNet	1.26	1.39	2.57	1.74	0.600	0.251
PoinTr	0.76	1.05	1.88	1.23	0.575	0.421
ProxyFormer	0.44	0.67	1.33	0.81	0.556	**0.466**
SeedFormer	0.48	0.70	1.30	0.83	0.561	0.452
SVDFormer	0.46	0.65	**1.13**	0.75	**0.538**	0.457
ODGNet	**0.44**	**0.64**	1.14	**0.74**	0.581	0.451
Ours	**0.44**	**0.63**	**1.13**	**0.73**	**0.539**	**0.468**

Methods	21 unseen categories					
	CD-S↓	CD-M↓	CD-H↓	CD-Avg↓	DCD-Avg↓	F1↑
PCN	3.17	3.08	5.29	3.85	0.644	0.101
GRNet	1.85	2.25	4.87	2.99	0.625	0.216
PoinTr	1.04	1.67	3.44	2.05	0.604	0.384
ProxyFormer	**0.60**	1.13	2.54	1.42	0.583	**0.415**
SeedFormer	0.61	1.07	2.35	1.34	0.586	0.402
SVDFormer	0.61	1.05	**2.19**	**1.28**	**0.554**	**0.427**
ODGNet	**0.59**	**1.01**	2.26	1.29	0.597	0.415
Ours	0.61	**1.03**	**2.16**	**1.27**	**0.558**	0.414

Table 3: The quantitative results on the ShapeNet-34 dataset.

Metrics/Methods	TopNet	PCN	GRNet	SeedFormer	ODGNet	**Ours**
FD($\times 10^3$)↓	5.354	2.235	0.816	0.151	1.28	**0.063**
MMD($\times 10^3$)↓	0.636	1.366	0.568	0.516	**0.299**	0.571

Table 4: The quantitative results on the KITTI dataset.

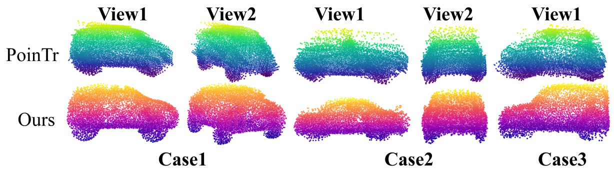

Figure 8: The visualization results on the KITTI dataset.

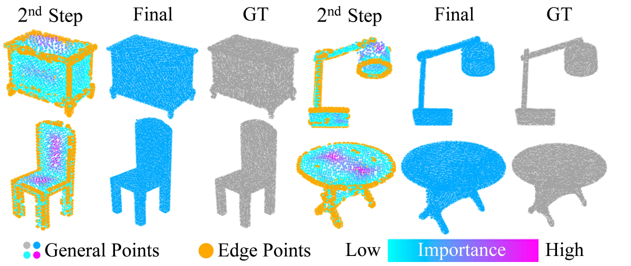

Figure 9: The visualization results of inverse edge-aware importance. The bigger orange points represent the edge points, while the smaller points represent the general points. The **magenta** represents higher importance, while the **cyan** represents the lower one.

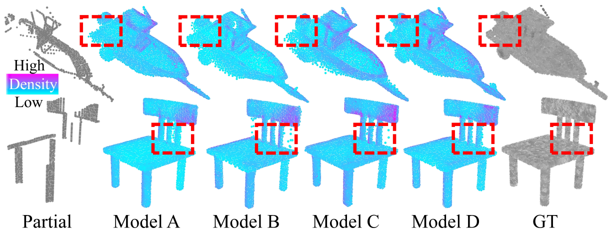

Figure 10: The visualization results of the ablation experiment for each module. The **magenta** represents higher density, while the **cyan** represents the lower one.

Model	IEup	PMMA		CD↓	DCD↓	F1↑
		Project	Scaled			
A	○	○	○	6.54	0.536	0.841
B	✓	○	○	6.51	0.532	0.843
C	✓	✓	○	6.72	0.547	0.829
D	✓	✓	✓	**6.48**	**0.532**	**0.844**

Table 5: The ablation study of each component.

Methods	Params	FLOPs	CD↓	DCD↓	F1↑
GRNet	76.71M	25.88G	8.83	0.622	0.708
PoinTr	31.28M	**10.60G**	8.38	0.611	0.745
SeedFormer	**3.31M**	53.76G	6.74	0.583	0.818
SVDFormer	32.63M	39.26G	6.54	0.536	0.841
Ours	60.88M	75.86G	**6.48**	**0.532**	**0.844**

Table 6: The parameters and FLOPs between different methods.

module can also extract the edge points effectively.

Ablation on PMMA Module. The Model B, C, and D in Table 5 verify the effectiveness of the PMMA module. Model C only adds a point-to-pixel feature projection module to Model B, which leads to worse results. This is because Model B directly extracts a set of image features from ResNet, while the image features in Model C are composed of $\mathcal{N}$ sets of features. The magnitude of the image features in Model C is much larger than that of Model B, resulting in the features not being able to function normally in the subsequent steps. For Model D, the scaling factor is added, which can shrink the groups of features extracted from Model C and restore the feature vectors to the normal magnitude. In Figure 10, we visualize the density of the point clouds generated by the four sets of models. Compared with Model D, Model C has a higher density of complete point clouds around the input points, such as the edge of the hull and the back of the chair, which indicates that unscaled features will affect the distribution of global features and lead to an imbalanced density of point clouds.

4.4 Model Complexity and Resource Costs

The detailed complexity and resource cost analyses are shown in Table 6. While the two-step alignment does incur additional computational costs and the EVP module requires more parameters, these design choices were carefully considered to achieve the demonstrated performance improvements, which reduce CD by 0.06×10^{-3} over the baseline.

5 Conclusion and Discussion

In this article, we propose IE-PMMA, which is the first point cloud completion method that takes into account the edge information of the point cloud. The PMMA module achieves precise point-to-pixel alignment, significantly improving upon previous coarse feature matching methods. The novel inverse weighting approach in the IEup module prioritizes global shape coherence to prevent local detail overfitting. The EPV module's probabilistic volume representation offers enhanced noise robustness and broader applicability. Various experiments show that IE-PMMA achieves the most advanced performance in the point cloud completion task.

Acknowledgments

This work was supported by the Sichuan Science and Technology Program under Grant 2025YFHZ0034. The authors would like to thank Zi Guo from the Nuclear Power Institute of China for the helpful and valuable discussions.

References

[Bazazian and Parés, 2021] Dena Bazazian and M Eulàlia Parés. Edc-net: Edge detection capsule network for 3d point clouds. *Applied Sciences*, 11(4):1833, 2021.

[Cai *et al.*, 2024] Pingping Cai, Deja Scott, Xiaoguang Li, and Song Wang. Orthogonal dictionary guided shape completion network for point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 864–872, 2024.

[Chang *et al.*, 2015] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[Chen *et al.*, 2017] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovačević. Fast resampling of three-dimensional point clouds via graphs. *IEEE Transactions on Signal Processing*, 66(3):666–681, 2017.

[Chen *et al.*, 2022] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Deformable feature aggregation for dynamic multi-modal 3d object detection. In *European conference on computer vision*, pages 628–644. Springer, 2022.

[Chen *et al.*, 2023] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13581–13590, 2023.

[Choy *et al.*, 2016] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016.

[Deng *et al.*, 2022] Qinwen Deng, Songyang Zhang, and Zhi Ding. An efficient hypergraph approach to robust point cloud resampling. *IEEE Transactions on Image Processing*, 31:1924–1937, 2022.

[Duan *et al.*, 2025] Fan Duan, Jiahao Yu, and Li Chen. T-corresnet: template guided 3d point cloud completion with correspondence pooling query generation strategy. In *European Conference on Computer Vision*, pages 90–106. Springer, 2025.

[Geiger *et al.*, 2013] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[Gumhold *et al.*, 2001] Stefan Gumhold, Xinlong Wang, Rob S MacLeod, et al. Feature extraction from point clouds. In *IMR*, pages 293–305, 2001.

[Hackel *et al.*, 2016] Timo Hackel, Jan D Wegner, and Konrad Schindler. Contour detection in unstructured 3d point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1610–1618, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Himeur *et al.*, 2021] Chems-Eddine Himeur, Thibault Lejemble, Thomas Pellegrini, Mathias Paulin, Loic Barthe, and Nicolas Mellado. Pcednet: A lightweight neural network for fast and interactive edge detection in 3d point clouds. *ACM Transactions on Graphics (TOG)*, 41(1):1–21, 2021.

[Huang *et al.*, 2024] Shangfeng Huang, Ruisheng Wang, Bo Guo, and Hongxin Yang. Pbwr: Parametric-building-wireframe reconstruction from aerial lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27778–27787, 2024.

[Jaklič *et al.*, 2015] Aleš Jaklič, Miran Erič, Igor Mihajlović, Žiga Stopinšek, and Franc Solina. Volumetric models from 3d point clouds: The case study of sarcophagi cargo from a 2nd/3rd century ad roman shipwreck near sutivan on island brač, croatia. *Journal of Archaeological Science*, 62:143–152, 2015.

[Jia *et al.*, 2024] Ran Jia, Junpeng Xue, Wenbo Lu, Zeyu Song, Zhichao Xu, and Shuxin Lu. A coupled calibration method for dual cameras-projector system with sub-pixel accuracy feature extraction. *Sensors*, 24(6):1987, 2024.

[Kim *et al.*, 2019] Hak Gu Kim, Heoun-Taek Lim, and Yong Man Ro. Deep virtual reality image quality assessment with human perception guider for omnidirectional image. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(4):917–928, 2019.

[Koch *et al.*, 2019] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Li *et al.*, 2023] Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9466–9475, 2023.

[Liu *et al.*, 2022] Hao Liu, Hui Yuan, Junhui Hou, Raouf Hamzaoui, and Wei Gao. Pufa-gan: A frequency-aware generative adversarial network for 3d point cloud upsampling. *IEEE Transactions on Image Processing*, 31:7389–7402, 2022.

[Ma *et al.*, 2022] Xuehan Ma, Xueyan Li, and Junfeng Song. Point cloud completion network applied to vehicle data. *Sensors*, 22(19):7346, 2022.

[Matveev *et al.*, 2022] Albert Matveev, Ruslan Rakhimov, Alexey Artemov, Gleb Bobrovskikh, Vage Egiazarian, Emil Bogomolov, Daniele Panozzo, Denis Zorin, and Evgeny Burnaev. Def: Deep estimation of sharp geometric features in 3d shapes. *ACM Transactions on Graphics*, 41(4), 2022.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[Tesema *et al.*, 2023] Keneni W Tesema, Lyndon Hill, Mark W Jones, Muneeb I Ahmad, and Gary KL Tam. Point cloud completion: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[Vaswani, 2017] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[Wang *et al.*, 2019] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[Wang *et al.*, 2020] Xiaogang Wang, Yuelang Xu, Kai Xu, Andrea Tagliasacchi, Bin Zhou, Ali Mahdavi-Amiri, and Hao Zhang. Pie-net: Parametric inference of point cloud edges. *Advances in neural information processing systems*, 33:20167–20178, 2020.

[Weber *et al.*, 2010] Christopher Weber, Stefanie Hahmann, and Hans Hagen. Sharp feature detection in point clouds. In *2010 shape modeling international conference*, pages 175–186. IEEE, 2010.

[Wen *et al.*, 2023] Cheng Wen, Baosheng Yu, and Dacheng Tao. Learnable skeleton-aware 3d point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17671–17681, 2023.

[Wu *et al.*, 2023] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2023.

[Xiang *et al.*, 2021] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5499–5509, 2021.

[Yang *et al.*, 2018] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.

[Ye *et al.*, 2023] Yunfan Ye, Renjiao Yi, Zhirui Gao, Chenyang Zhu, Zhiping Cai, and Kai Xu. Nef: Neural edge fields for 3d parametric curve reconstruction from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8486–8495, 2023.

[Yu *et al.*, 2018] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018.

[Yu *et al.*, 2021] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.

[Yu *et al.*, 2024] Jinpeng Yu, Binbin Huang, Yuxuan Zhang, Huaxia Li, Xu Tang, and Shenghua Gao. Geoformer: Learning point cloud completion with tri-plane integrated transformer. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8952–8961, 2024.

[Yuan *et al.*, 2018] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018.

[Zhang *et al.*, 2021] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15890–15899, 2021.

[Zhou *et al.*, 2022] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision*, pages 416–432. Springer, 2022.

[Zhu *et al.*, 2023a] Xiangyu Zhu, Dong Du, Weikai Chen, Zhiyou Zhao, Yinyu Nie, and Xiaoguang Han. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13601–13610, 2023.

[Zhu *et al.*, 2023b] Zhe Zhu, Honghua Chen, Xing He, Weiming Wang, Jing Qin, and Mingqiang Wei. Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14508–14518, 2023.

[Zhu *et al.*, 2023c] Zhe Zhu, Liangliang Nan, Haoran Xie, Honghua Chen, Jun Wang, Mingqiang Wei, and Jing Qin. Csdn: Cross-modal shape-transfer dual-refinement network for point cloud completion. *IEEE Transactions on Visualization and Computer Graphics*, 2023.