Current Science
CS 4 (6), 903– 911 (2024)
Received 24 Sept 2024 | **Revised** 25 Oct 2024 | **Accepted** 26 Nov. 2024 | **Online Available** 10 Dec 2024
https://doi.org/10.52845/CS/2024-4-6-13

Current Science

Current Science

OPEN ACCESS

ISSN(O)2795-8639

ORIGINAL ARTICLE

Dynamic Graph Structure-Based Point Cloud Complementation for Real Underwater Scene

Zuoqi Hu[1], Mengfan Zhang[2], Zhe Song[3], Wenhan Wang[4], Hui Yu[5], Ting Wang[6], Qing Cai[7], Shu Zhang[8]

[1]Institution: School of Computer Science and Technology, Ocean University of China

[2]Institution: School of Computer Science and Technology, Ocean University of China

[3]Institution: School of Computer Science and Technology, Ocean University of China

[4]Institution: School of Computer Science and Technology, Ocean University of China

[5]Institution: University of Portsmouth

[6]Institution: Shandong University of Science and Technology

[7]Institution: School of Computer Science and Technology, Ocean University of China

[8]Institution: School of Computer Science and Technology, Ocean University of China

*Corresponding Author: Shu Zhang

Abstract

In numerous applications like underwater exploration, object recognition, and navigation, the completion of point clouds holds significant importance. Those point clouds are commonly generated by underwater cameras with multi-view stereoscopy. However, the dramatic negative impact of the underwater environment on visual data leads to a significant drop in the quality of those point clouds, such as noise and incomplete 3D data. To tackle these challenges, this study introduces an innovative network for refining and completing underwater point clouds, named UDPCN. The network is designed to forecast and fill in the missing areas within point clouds in an underwater setting. Through the point cloud optimization module, the proposed technique learns the underlying flow shape of the pristine point cloud and captures the distribution of the subpar point cloud using a score-based model. The global optimization is then conducted by applying the predicted displacement of each point to the origin in the proposed pipeline. We integrate an improved DGCNN into the encoder to perform local feature extraction. Subsequently, the proposed method introduces a geometric perception encoder to extract global features of the point cloud, enabling the comprehensive modeling of structural information and inter-point relationships within point clouds. In order to demonstrate the effectiveness of the proposed approach, experiments and evaluations are conducted with an underwater real-scene dataset constructed by this paper, which is achieved by collecting CAD data using 3D scanners and underwater cameras. The superiority of the proposed method's performance over several existing methods in real underwater scenes is substantiated through extensive experiments and comparisons.
Keywords: Point Cloud Completion; Optimization; Underwater

Introduction

Refining and completing underwater point clouds is crucial in fields like underwater robotics and computer vision. However, there are significant challenges due to sensor limitations, water turbidity, and occlusions. The noise distribution in underwater point clouds is affected by impurity interference, and data often goes missing. Existing methods (Zhang et al. 2020; An et al. 2024) are not tailored for the underwater environment and struggle to handle these issues.

Early researchers used traditional filtering methods like Gaussian and low-pass filtering to reduce point cloud drift caused by airborne impurities. But these methods are ineffective for underwater impurities due to their more complex distribution. Recently, deep learning techniques (Charles et al. 2017; Qi et al. 2017) have shown good results with neural networks for point clouds. Luo et al. proposed a framework for optimizing point cloud reconstructions. Their method learns the manifold of noisy point clouds and filters high-noise points in downsampling, but this causes detail loss. Thus, existing methods are not suitable for underwater point clouds.

Besides point cloud shifts from underwater impurities, underwater environments often cause missing points. Initial point cloud completion efforts explored voxelization and 3D convolution, but they were computationally complex and costly. The breakthroughs of PointNet and PointNet++ (Charles et al. 2017; Qi et al. 2017) introduced a mainstream way of using 2D convolutional neural networks for 3D point clouds. However, these methods have challenges in accurately recovering complex point cloud structures due to frequent max-pooling in the encoding stage.

In this research paper, we introduce an innovative network named UDPCN for underwater real-world scene point cloud completion. This model improves the accuracy and fusion of local and global features for joint prediction, enhancing the efficiency. As shown in Figure 1, in the proposed network architecture, we focus on enhancing feature extraction during the encoding phase.

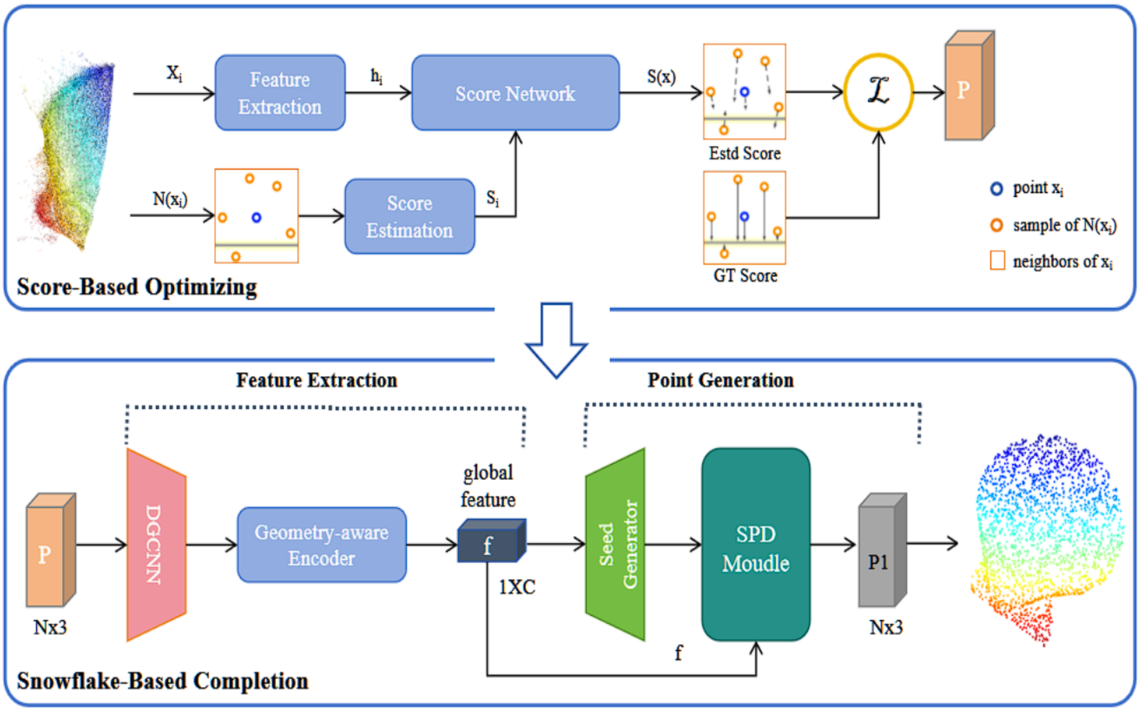


Figure 1: Our UDPCN network consists of an initial preprocessing stage, a feature extraction module, and a point generation module

The network is tailored to preprocess underwater point clouds, eliminating impurity points and regularizing the distribution. To address the challenge of insufficient local details in reconstruction due to the underwater environment, we propose an approach combining a lightweight DGCNN (Phan et al. 2018) and a geometric perception encoder (Yu et al. 2021). These components guide the completion process. We assessed UDPCN using our Shell dataset and the public PCN dataset (Yuan et al. 2018) for comparison. The outcomes show the excellence of our approach in handling real-world scenes.

Our primary contributions are outlined below.

- A customized optimizing network for preprocessing real underwater point clouds, optimizing them by learning implicit gradient fields.

- A novel lightweight DGCNN is proposed based on dynamic graphs, which effectively integrating point relationships, our approach significantly enhances the preservation of input point cloud fine-grain information.
- A new underwater point cloud dataset regarding underwater shells. This dataset provides paired ground truth and partial data for training underwater scene-based point cloud completion models.

2. Related Work

2.1 Point Cloud Completion

A subset of 3D reconstruction is completing shapes from point clouds. Scholars increasingly use unstructured point clouds to represent 3D objects for efficient memory use and detail capture. PointNet and its variants (Charles et al. 2017; Qi et al. 2017) directly process 3D coordinates and inspire research. The PCN method (Yuan et al. 2018) presented a completion framework. Subsequent works explored local or multi-scale features. While these methods complete point clouds in virtual and aerial datasets, they are not suitable for underwater due to low light and scattering interference. Thus, underwater point cloud details are not well-captured, limiting their application.

2.2 Point Cloud Optimizing

The emergence of neural networks (Charles et al. 2017; Qi et al. 2017) based on point clouds has enabled profound optimization of 3D data structures. PointCleanNet (Rakotosaona et al. 2019) uses a subset of PointNet as its base. TotalDn (Casajus, Ritschel, and Ropinski 2019) offers an unsupervised framework for optimizing point clouds and deriving loss functions for training. Score Matching is a method for training energy models, minimizing squared differences. It's been used for 3D models. While these methods handle noise in airborne point cloud generation, underwater point cloud generation has more challenges. Natural noise isn't the only issue underwater; impurities and light scattering add complexities that the above models can't handle well.

3. Methodology

3.1 Network Architecture

In our approach, we start by using an optimization network to enhance the underwater point cloud. This network learns the distribution relationship of the underwater point cloud and GT and optimizes the initial point cloud. Then, the optimized point cloud undergoes feature extraction for encoding local and non-local geometric features and is used in a scoring estimation module to create a scoring function. The features and scores are used to construct a gradient field for point cloud optimization, which is input to the completion network. First, a DGCNN extracts local features that are encoded by a geometric perception encoder. A seed generation module creates a coarse point cloud capturing the complete representation. Finally, the SPD network with multiple deconvolution layers and Skip-Transformers is used for point splitting to ensure consistency and the final refined point cloud is output.

3.2 Score-Based Optimizing

A score-based optimization approach is used to refine underwater point clouds. It focuses on learning the distribution of clean point clouds, modeling impurity noise, and capturing the discrete distribution near currents through a gradient field. Based on gradient ascent, it aims to remove noisy points and cluster discrete points before completion.

The score-estimating network is crucial. Given an impurity point cloud $\mathbf{P} = \{p_i\}_{i=1}^{N}$, the network predicts gradients of the logarithmic probability function for each point. These units extract multi-scale and non-local features via a dynamic graph convolutional layer stack and enhance contextual information through dense connections.

The score estimation units are parameterized by the features h_i of point p_i. The score estimation units are formally defined as follows:

$$Si(p) = Score(p - p_i, h_i),$$

Score$(\cdot)$ is a MLP.

We represent impurity-free point cloud as $T = t_{i=1}^{N}$. We define the score for a point $p \in \mathbb{R}^3$ with respect to the ground truth t as follows:

$$s(p) = N(p, T) - p, p \in R^3,$$

where the closest point in T to p is $N(p,T)$.

The training goal is to make the predicted values of the network match the ground truth values defined above:

$$\mathcal{L}_i = E_{x\sim N(p_i)}\,[\| s(p) - S_i(p) \|_2^2],$$

Where $N(p_i)$ is a distribution centered around the neighborhood of p_i.

The aggregation of the individual score function goals for each local region is simply the final training goal: $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} L^i$.

3.3 Snowflake-Based Completion

DGCNN Module:

We present an improved DGCNN with enhanced feature extraction capabilities. Our method integrates a Transformer into the DGCNN for feature extraction and passes each feature through skip merge.

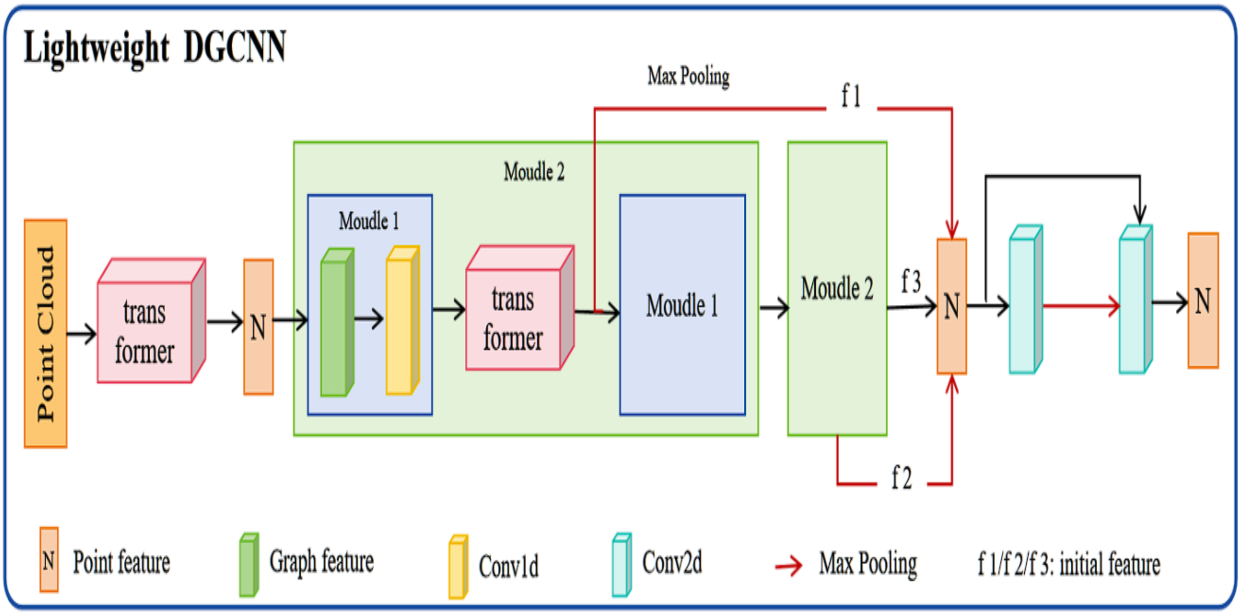


Figure 2: The whole of the improved DGCNN

Figure 2 shows the key components of the improved DGCNN. To improve the representation of input points, we use a transformer to map them to a higher-dimensional space. Then, a graph convolutional feature extractor and a 2D convolutional layer conduct the initial feature extraction f_1. The resulting features are maximally pooled and sent to the final feature aggregation network. Meanwhile, we upscale the extracted features to a higher dimension. Finally, we extract higher-level features f_2 using the graph convolutional feature extractor and 2D convolutional layer, and aggregate them with the initially extracted features.

Geometry-aware Encoder:

We use an encoder to transform the obtained point cloud and relational features into a high-dimensional space, as shown in Figure 3.

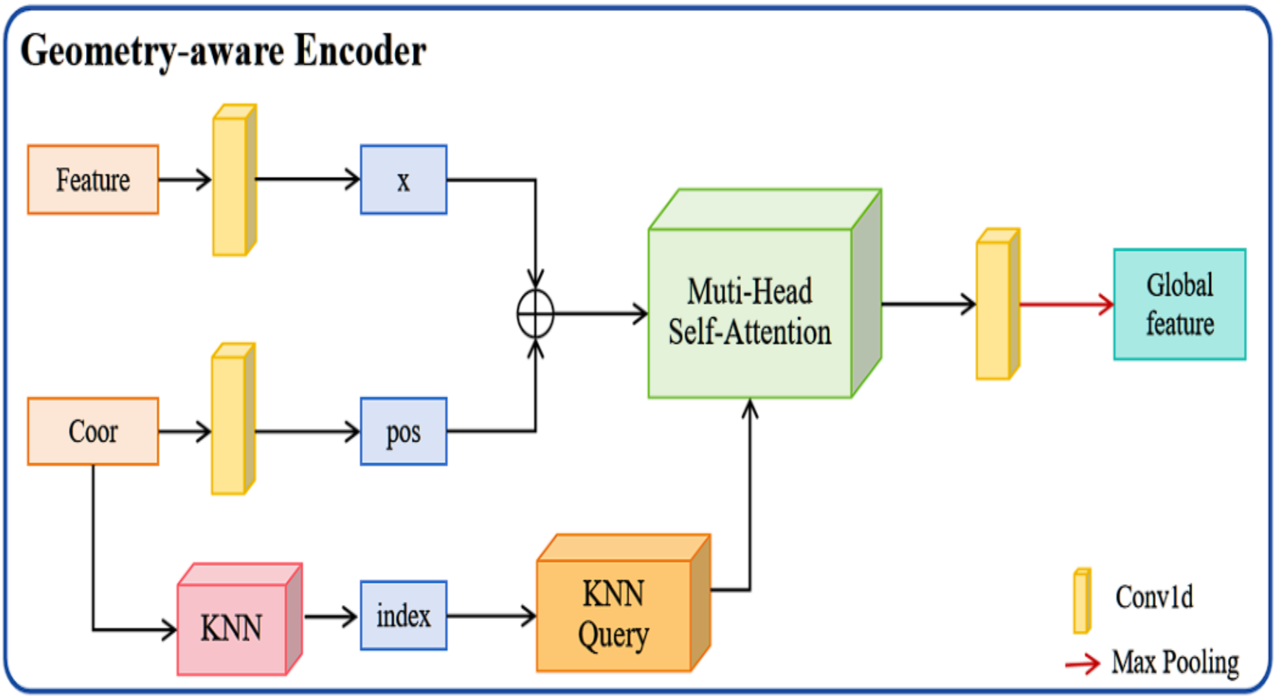


Figure 3: The overall of the Geometry-aware encoder

We process the point coordinates ($coor$) and features f acquired from DGCNN. Initially, we use the get_knn_index module to get the coordinates of nearby points in the point cloud. At the same time, we encode the point's location (pos) with a convolutional layer. The resulting features are fused with the basic features obtained through knn queries near the point. Moreover, we use a $multi-headself-attention$ mechanism to capture both local and global geometric features. Finally, convolution and pooling operations are applied to aggregate the local features into global features.

Training Loss:

The evaluation of point cloud completion quality heavily relies on the choice of the loss function. In our research, we opted for chamfer distance(CD) (Hajdu, Hajdu, and Tijdeman 2012) as the loss function. Chamfer distance quantifies the dissimilarity between two sets of 3D point clouds, denoted as:

$$d_{CD}(P_A, P_B) = \frac{1}{P_A}\sum_{x\in P_A} \| x - y \|_2^2 + \frac{1}{P_B}\sum_{x\in P_B} \| y - x \|_2^2$$

Where P_A and P_B are two point clouds for comparison.

4. Experimental and Analysis

4.1 Dataset

PCN:

The proposed Shell dataset comes from shell models scanned in the underwater environment, which are natural. The shells were carefully selected for having many similar features and significant differences. They produce point cloud data underwater, affected by impurities. Poor underwater lighting limits point cloud detail information, adding challenges. Figure 4 shows real scan samples and 3D shapes, visualizing the data's challenges and characteristics.

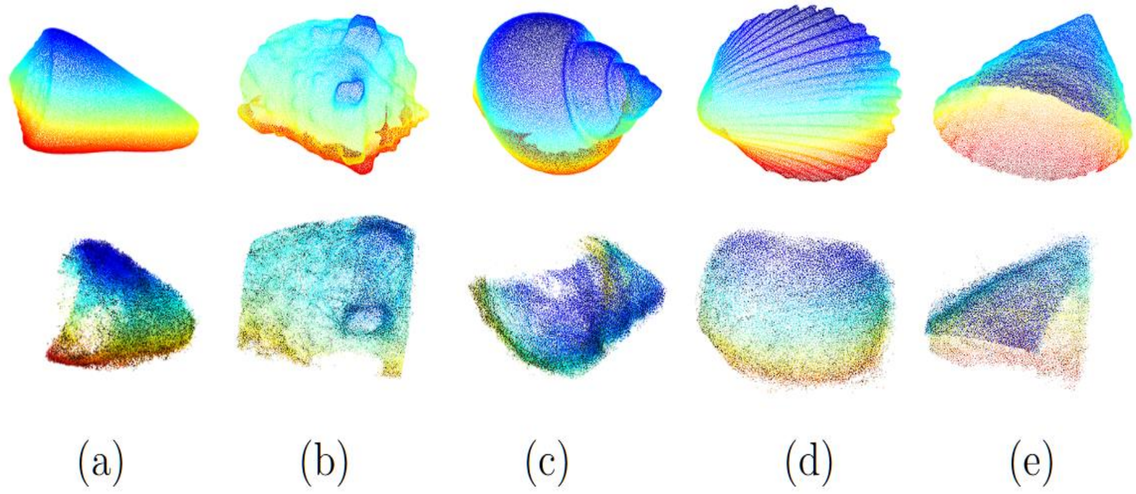


Figure 4: An instance of real-world underwater 3D point clouds

Shell:

The proposed Shell dataset comes from shell models scanned in the underwater environment, which are natural. The shells were carefully selected for having many similar features and significant differences. They produce point cloud data underwater, affected by impurities. Poor underwater lighting limits point cloud detail information, adding challenges. Figure 4 shows real scan samples and 3D shapes, visualizing the data's challenges and characteristics.

4.2 Implementation Details

In evaluating shape completion quality, we use two main metrics: Chamfer Distance (CD) (Hajdu, Hajdu, and Tijdeman 2012) and F-score (Piotroski 2000). CD quantifies the distance between the completed surface and GT; lower values mean higher accuracy. F-score is a comprehensive metric considering precision and recall; higher values indicate better completion quality.

4.3 Quantitative Results

In Table 1, we compared various methods using an 8-category dataset. All methods were trained on 8 training sets simultaneously and evaluated on the PCN test set. Our UDPCN model showed competitive performance in average L1 Chamfer Distance and F-score metrics compared to the current advanced methods, achieving the highest performance.

Table 1: The evaluation of techniques on our dataset is compared using the Chamfer Distance ($cd - l1 \times 10^2$), Lower values indicate better results. Additionally, we consider the F-score ($F - score@1\% \times 10^2$) where higher values are preferable.

	Chamfer Distance l1 $\times 10^2$									F-Score↑
	Air	Cab	Car	Cha	Lam	Sof	Tab	Wat	Overall↓	
FoldingNet	11.91	19.38	15.40	21.19	20.99	20.55	17.59	16.58	17.95	0.25
TopNet	9.91	17.47	14.47	18.34	17.50	19.61	15.16	14.83	15.91	0.32
PCN	10.04	20.32	15.51	19.30	18.42	21.13	17.01	15.03	17.09	0.31
PoinTr	7.75	13.14	11.63	11.50	9.31	13.50	9.91	8.96	10.71	0.54
SnowFlake	7.62	9.84	8.67	8.82	8.98	10.19	7.40	7.56	8.63	0.72
USSPA	8.56	9.71	9.35	8.40	10.63	15.47	7.84	7.93	9.73	0.63
Ours	**7.03**	**8.93**	**8.05**	**8.17**	**7.55**	**9.32**	**7.19**	**6.84**	**7.88**	**0.78**

Table 2 shows the results of our method for underwater real-scene restoration tasks using the Shell dataset. The results demonstrated the excellent performance of our network in completing restoration tasks in noisy underwater real-scene environments. Specifically, our method achieved an average improvement of 44.6% in $cd - l1$ and a significant 55.4% improvement in $F - score$ compared to existing SOTA methods.

4.4 Qualitative Results

Table 2: The results of the comparison made using our Shell dataset.

	FoldingNet	TopNet	PCN	PoinTr	USSPA	SnowFlake	Ours
F-Score↑	0.139	0.125	0.131	0.172	0.332	0.367	**0.516**
CD-L1↓	95.32	107.45	96.05	57.718	27.350	26.69	**12.19**

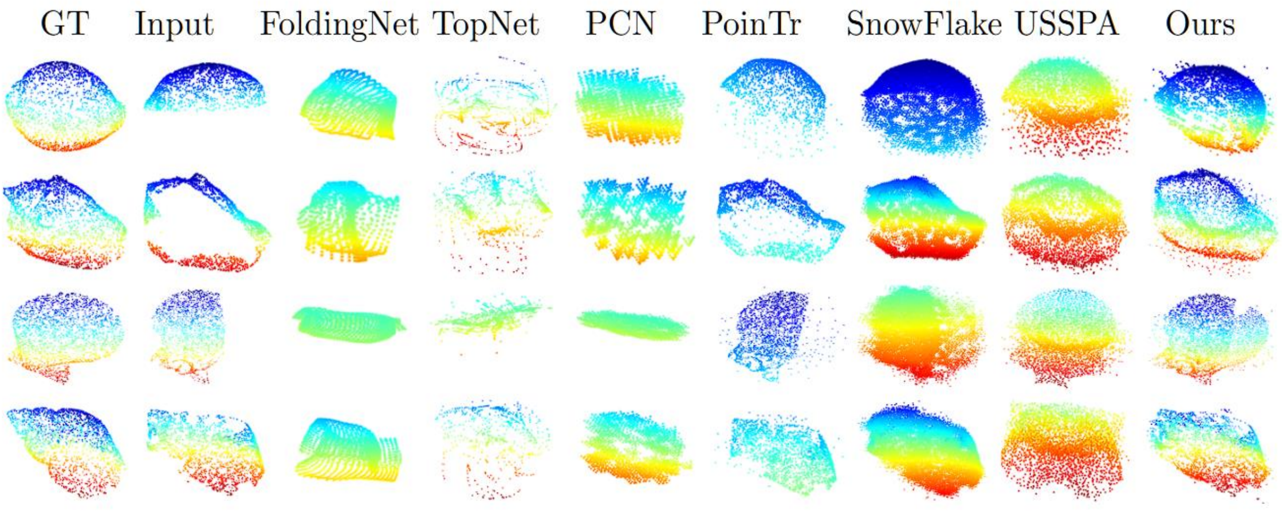


Figure 5: All methods were evaluated qualitatively on our Shell dataset.

Figure 5 shows the qualitative results of the eight evaluated methods. Our UDPCN performs well in handling incomplete point clouds. It recovers the basic shape and adds details for more accurate and realistic completions. For example, in the third data, our results predict the basic geometry, while others fail and have scattered points. These prove our method's superiority. It surpasses previous ones, showing its effectiveness in high-quality completions.

4.5 Ablation Study

To evaluate the validity of different architectural designs in our approaches, we conducted ablation experiments. We selectively removed specific modules from our complete UDPCN and compared the results with those of the ablated models. The results in Table 3 clearly highlight the significance of these modules in improving the accuracy of point cloud completion. The inclusion of these modules in our full model greatly contributes to the overall performance improvement of our approach.

Table 3: Effectiveness of DGCNN and Geometry-aware Encoder modules in PCN datasets

Chamfer Distance l1 ×102										
DGCNN	**Geo**	**Air**	**Car**	**Cha**	**sof**	**Lam**	**Cab**	**Tab**	**Wat**	**Overall↓**
×	×	7.75	8.67	8.82	10.19	8.98	9.84	7.40	7.56	8.66
✓	×	7.32	8.27	8.71	9.95	8.67	9.61	7.39	7.26	8.40
×	✓	7.33	8.31	8.55	9.67	8.35	9.56	7.33	7.08	8.27
✓	✓	**7.03**	**8.05**	**8.17**	**9.32**	**7.55**	**8.93**	**7.19**	**6.84**	**7.88**

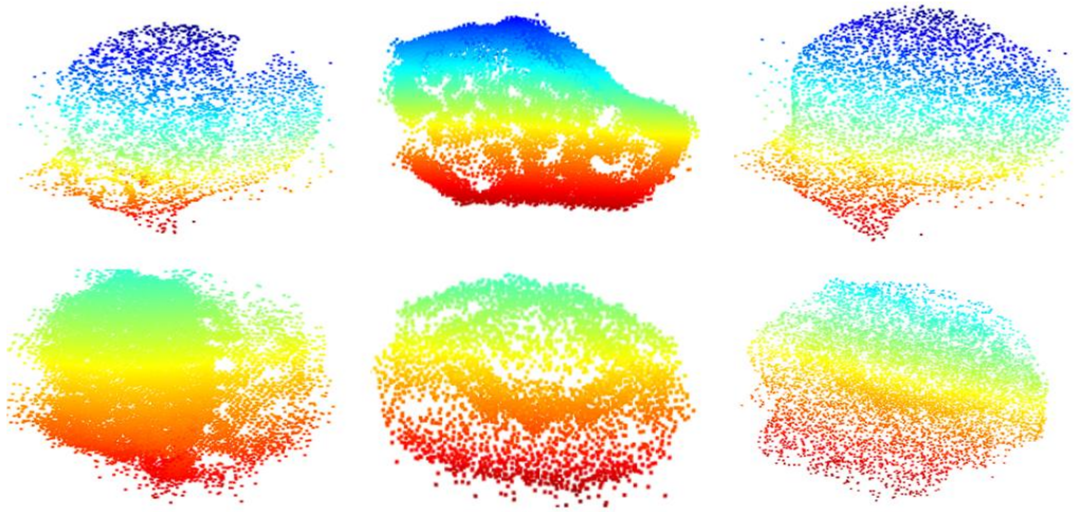
To assess the efficacy of different architectural designs in our network. First, we added the geometric perception encoder to the baseline model for global features. Second, we removed the DGCNN network for local feature extraction. Also, we annotated the entire network including the encoder and DGCNN. From Table 3, the complete network outperforms all other variations. This comparison shows the validity of using both for better results.

Our optimizing module fits planes to approximate discrete points in the point cloud instead of simply removing them for noise reduction.

Table 4: The impact of the optimization module on our Shell dataset

Optimized module	CD-$l1$ ↓	F-Score↑
×	27.53	0.328
✓	**12.19**	**0.516**

Table 4 shows that our module is good at keeping more point cloud details, improving the completion task. In Figure 6 (a), our module removes most noise while keeping global and local features. It doesn't discard any points, showing its effectiveness in maintaining key information for accurate completion.



(a) (b) (c)

Figure 6: Comparison of full networ(first row) and without the optimizing module(second row)

Discussion and Conclusions

We introduced UDPCN for completing real underwater scene point clouds. Our approach optimizes the input through the underlying manifold of the clean point cloud and uses a score-based model to adjust point displacements for better results. A key contribution is the geometry-aware module that combines local and global features for improved prediction accuracy, speed, and completeness. We also created a new underwater point cloud dataset of 55 shell models with paired real and partial data for underwater scene training.

Although UDPCN has progressed, there are limitations. One is that completed results may lack details compared to the ground truth, as the current method focuses on global features and underwater point cloud localization. Improving the representation of local features could improve the quality of completion.

Acknowlegements

This work was supported by the Natural Science Foundation of Shandong Province of China, the Hainan Provincial Joint Project of Sanya Yazhou Bay Science and Technology City, the Natural Science Foundation of China, the National Key R&D Program of China under Grant,China Ministry of Education's' Chunhui Program on International Collaborative Research.

References:

1. An, K., Fan, H., & Dong, J. (2024). "A marker-based method for visual-inertial initialization," Intelligent Marine Technology and Systems, 2(1), 27.
2. Casajus, Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski (2019), "Total denoising: Unsupervised learning of 3D point cloud cleaning," in *2019 IEEE/CVF international conference on computer vision (ICCV)*.
3. Charles, R. Qi, Hao Su, Mo Kaichun, and Leonidas J. Guibas (2017), "PointNet: Deep

learning on point sets for 3D classification and segmentation," in *2017 IEEE conference on computer vision and pattern recognition (CV PR)*.
4. Hajdu, András, Lajos Hajdu, and Robert Tijdeman (2012), "Approximation of the euclidean distance by chamfer distances," *Acta Cybernetica*, 20 (3), 399–417.
5. Phan, Anh Viet, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui (2018), "DGCNN: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, 533–43.
6. Piotroski, Joseph D. (2000), "Value investing: The use of historical financial statement information to separate winners from losers," *Journal of Accounting Research*, 38, 1.
7. Qi, CharlesR., Li Yi, Hao Su, and LeonidasJ. Guibas (2017), "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Cornell University - arXiv, Cornell University - arXiv*.
8. Rakotosaona, Marie-Julie, VittorioLa Barbera, Paul Guerrero, NiloyJ. Mitra, and Maks Ovsjanikov (2019), "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," *arXiv: Graphics,arXiv: Graphics*.
9. Yu, Xumin, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou (2021), "PoinTr: Diverse point cloud completion with geometry-aware transformers," in *2021 IEEE/CVF international conference on computer vision (ICCV)*.
10. Yuan, Wentao, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert (2018), "PCN: Point completion network," in *2018 international conference on 3D vision (3DV)*.
11. Zhang, Shu, Hui Yu, Ting Wang, and Junyu Dong (2020), "Augmented visual feature modeling for matching in low-visibility based on cycle-labeling of superpixel flow," *Knowledge-Based Systems*, 195, 105699.