PointCFormer: A Relation-Based Progressive Feature Extraction Network for Point Cloud Completion

Yi Zhong[1], Weize Quan[2,3], Dong-Ming Yan[2,3], Jie Jiang[1*], Yingmei Wei[1]

[1]National University of Defense Technology
[2]MAIS, Institute of Automation, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
{zhongyi, jiejiang, weiyingmei}@nudt.edu.cn, {qweizework, yandongming}@gmail.com

Abstract

Point cloud completion aims to reconstruct the complete 3D shape from incomplete point clouds, and it is crucial for tasks such as 3D object detection and segmentation. Despite the continuous advances in point cloud analysis techniques, feature extraction methods are still confronted with apparent limitations. The sparse sampling of point clouds, used as inputs in most methods, often results in a certain loss of global structure information. Meanwhile, traditional local feature extraction methods usually struggle to capture the intricate geometric details. To overcome these drawbacks, we introduce PointCFormer, a transformer framework optimized for robust global retention and precise local detail capture in point cloud completion. This framework embraces several key advantages. First, we propose a relation-based local feature extraction method to perceive local delicate geometry characteristics. This approach establishes a fine-grained relationship metric between the target point and its k-nearest neighbors, quantifying each neighboring point's contribution to the target point's local features. Secondly, we introduce a progressive feature extractor that integrates our local feature perception method with self-attention. Starting with a denser sampling of points as input, it iteratively queries long-distance global dependencies and local neighborhood relationships. This extractor maintains enhanced global structure and refined local details, without generating substantial computational overhead. Additionally, we develop a correction module after generating point proxies in the latent space to reintroduce denser information from the input points, enhancing the representation capability of the point proxies. PointCFormer demonstrates state-of-the-art performance on several widely used benchmarks.

Code — https://github.com/Zyyyyy0926/PointCFormer_Plus_Pytorch

Extended version — https://arxiv.org/abs/2412.08421

Introduction

Point cloud is a common data format in 3D vision tasks, and point cloud-based analysis tasks have greatly promoted the development of 3D computer vision (Mao et al. 2024). Point cloud data is generally obtained through 3D sensors (Bi and Wang 2010), however, due to inevitable external factors such as occlusion, light reflection, and limited sensor resolution (Chen and Yang 2016), as well as internal factors such as the inherent shape deficiencies of the object itself, point cloud data often exhibits incomplete and sparse characteristics. Point cloud completion is restoring this incomplete and sparse 3D point cloud data to a point cloud representing the complete shape of the object.

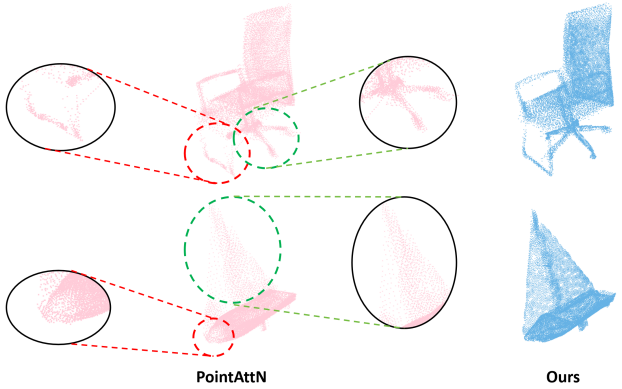

Figure 1: Improved global shape and enriched local detail in point cloud completion using our method.

Recent deep learning advances have solidified the encoder-decoder framework as a staple for point cloud completion, with PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b) laying the foundation for the feature encoder. The pioneering PointTr (Yu et al. 2021) leverages the Transformer model to enhance this architecture. SeedFormer (Zhou et al. 2022) introduces an innovative shape representation for feature integration, while (Wang et al. 2022) focuses on local feature grouping to improve completion. FBNet (Yan et al. 2022) combines feedback loops and cross transformers to better link feature levels, while SnowflakeNet (Xiang et al. 2021) employs skip transformers to inject spatial relationships into the decoding stage. AnchorFormer (Chen et al. 2023) introduces anchors for region differentiation and predicts the complete shape by merging these anchors with the observed input points. AdaPoinTr (Yu et al. 2023) improves the performance based on PointTr, while the Mamba structure in 3DMambaCom-

*Corresponding author: Jie Jiang

plete (Li, Yang, and Fei 2024) shows advantages in this field and pushes the envelope in point cloud completion.

Among these existing methods, the feature extraction of points is an essential step, which primarily relies on the combined use of down-sampling and k-nearest neighbor (k-NN) search. However, the risk of losing certain global structural information is inevitable if an overly sparse down-sampled point cloud is employed as the model input. In addition, traditional feature extraction algorithms based on simple k-NN search can result in the network only extracting relatively coarse local features. Unfortunately, these fundamental limitations have seldom been discussed in previous works.

In this paper, we delve into the utilization of an advanced feature extractor for point cloud completion, aiming to mitigate the loss of global information during point cloud sampling and the coarse extraction issues in local feature extraction. We propose a framework, termed PointCFormer, as shown in Fig. 2. Specifically, PointCFormer adopts the transformer-based encoder-decoder architecture. To handle the coarse extraction problem, we propose fine-grained computation for the contribution of the neighboring point to the target point according to the geometry characteristics of the local region (including 3D spatial coordinates and high-dimensional feature vectors). To better preserve the global shape information during the sampling process, we introduce a progressive method with slow down-sampling, which gradually refines the features through alternately global and local queries. Compared to previous work, Fig. 1 shows the superior performance of our method in the point cloud completion task in terms of global shape and local details. In summary, our main contributions are as follows:

- We introduce a *local geometric relationship perception* module, which adaptively determines point contribution weights within neighborhoods based on relation metrics in 3D space and high-dimensional feature space, thereby enhancing local feature extraction accuracy.
- We propose a *progressive feature extractor* that synergistically combines the advantages of kNN-based local perception and self-attention mechanism. The features are progressively refined through repeated alternation between global and local queries.
- We devise a *point proxy correction* module in the latent space to enhance the affinity between the generated point proxies and the original input point cloud, thereby ensuring that the generated proxies are more closely aligned with the input data.
- Our PointCFormer achieves state-of-the-art performance on four common datasets for point cloud completion.

Related Work

3D Point Cloud Completion

For 3D point cloud completion, traditional approaches (Dai, Ruizhongtai Qi, and Nießner 2017; Han et al. 2017; Stutz and Geiger 2018) employ voxelization or distance fields to describe 3D objects and process them using 3D convolutional neural networks (Wu et al. 2015). However, the high demand for memory and computational resources limits their application scope (Wang et al. 2022). To mitigate this issue, researchers have shifted towards using unstructured point clouds to represent 3D objects, exploring various innovative approaches. PointNet and its variants (Qi et al. 2017a,b) extract features directly from unstructured point clouds, provide a new perspective on the task of 3D point cloud processing. PCN (Yuan et al. 2018) leverages an encoder-decoder architecture and simulates the deformation process of the 2D plane through the FoldingNet technique (Yang et al. 2018). This maps 2D points onto 3D surfaces to achieve point cloud completion while preserving geometric structure and topological properties. SnowflakeNet (Xiang et al. 2021) models the point cloud generation process as a snowflake-like growth pattern based on certain base points in 3D space. This pattern gradually expands from the parent point to form a complete 3D shape, simulating complex geometric structures. LAKeNet (Tang et al. 2022) introduces a novel 3D shape prediction method, which effectively captures the key features of 3D shapes by considering the structured and topological information of the predicted 3D shapes and following a key point-skeleton-shape prediction process. PointAttN (Wang et al. 2024) transforms the traditional k-NN method of obtaining local features into an attention-based method. This refines the global dependency on the relationships between point clouds. SeedFormer (Zhou et al. 2022) introduces patch seeds as a new shape representation for point clouds and designs an upsampling transformer to make point cloud completion more efficient and accurate. Anchorformer (Chen et al. 2023) enhances the accuracy of point cloud completion by capturing region information through pattern recognition nodes and combining the information of sampling points and anchor points. Recently, (Li, Yang, and Fei 2024) applies the Mamba (Gu and Dao 2023) structure to point cloud completion, showcasing its advantages in processing 3D point cloud data, which brings new insights to the community.

Transformer

In the field of natural language processing, the Transformer architecture (Vaswani et al. 2017) has revolutionized sequence tasks with its self-attention and cross-attention mechanisms. In computer vision, Vision Transformer (Dosovitskiy et al. 2020; Han et al. 2022) applies the Transformer architecture to 2D image processing, successfully managing image classification by deconstructing images into token sequences. Further research, like DeiT (Touvron et al. 2021), has expanded the Transformer's efficient training strategy in visual tasks. However, applying the Transformer to 3D point cloud data is still exploratory (Guo et al. 2021; Wang 2023; Misra, Girdhar, and Joulin 2021). The unstructured, high-dimensional nature of such data presents certain challenges. Preliminary studies show that the Transformer can effectively capture local and global features in point clouds, providing a new approach to 3D point cloud recognition (Mao et al. 2021; Pan et al. 2021) and completion (Wen et al. 2022; Zhang et al. 2022; Fei et al. 2023). This involves using the Transformer's self-attention mechanism to handle point cloud data's local structure and the cross-attention mechanism to predict missing parts using existing point cloud information.

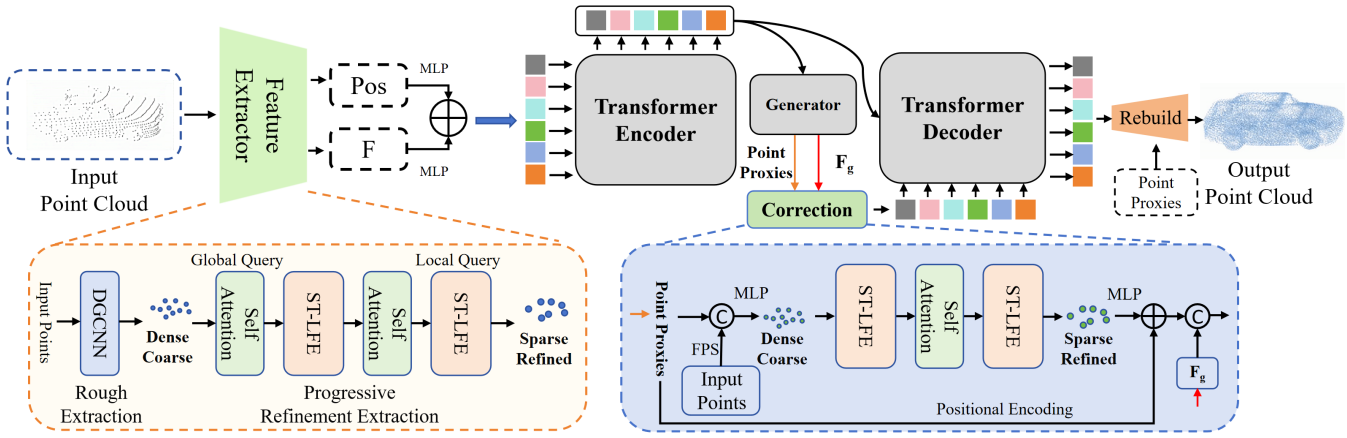

Figure 2: Overview of PointCFormer framework: Initially, we extract representative sampling points and their local features from the input incomplete point cloud using a feature extractor. After adding position embeddings to the local features, we employ a Transformer encoder-decoder architecture to predict point proxies for the missing parts. Concurrently, a correction module aligns these point proxies with the original point cloud distribution. Finally, a simple MLP and a Rebuild head are used to complete the point cloud based on the predicted point proxies in a coarse-to-fine manner.

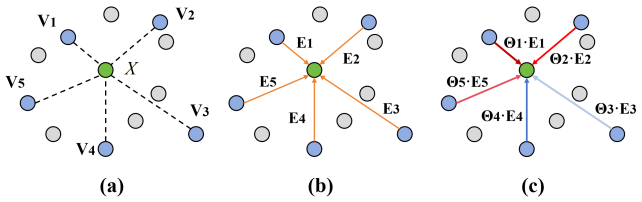

Figure 3: Comparison of local feature extraction: traditional kNN-based method (Left); DGCNN with relative relation fusion (Center); Ours (Right). In (c), the redder the line segment, the stronger the correlation between the two connected points; the bluer the line segment, the weaker the correlation.

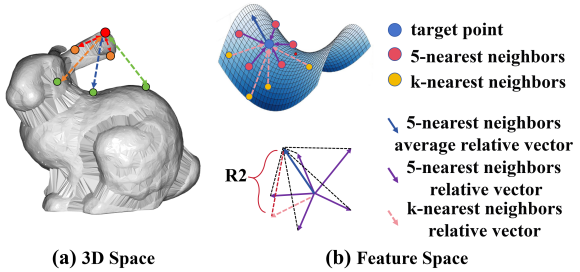

Figure 4: Quantification of the relationship between two points.

Proposed Method

Fig. 2 outlines the PointCFormer architecture for point cloud completion. The process begins by feeding an incomplete point cloud into a feature extractor, where a single-layer DGCNN performs initial dense sampling and extracts coarse features. These features are then subjected to a progressive refinement process. Employing a self-attention module to model global correlations and the scale-tailored local feature extractor (ST-LFE) to capture local detail features, the features transform from a dense and coarse state to a sparse and refined one. These features are then spatially position-encoded and fed into an encoder. In the latent space, the generator[1] uses the encoder's output features to create point proxies for the predicted 3D shape. Meanwhile, a correction module fuses these point proxies with spatial information from the dense input, ensuring alignment with the original point cloud distribution. Finally, the decoder outputs are used to gradually complete the point cloud with the help of an MLP and a reconstruction head.

[1]We follow the design of generator from (Yu et al. 2023).

Local Geometric Relationship Perception

Regular convolution operators (LeCun et al. 1998) on images cannot be directly applied to point cloud data. To address this challenge, many existing point cloud completion methods typically adopt a kNN-based method (Fig. 3(a)), which is essentially a variant of PointNet. This approach identifies local neighborhood points via k-NN search, maps these points' features to a high-dimensional space using an MLP, and then employs maximum pooling to extract local features. During the aggregation process, however, the relative positional relationships between points may be overlooked, thus limiting its effectiveness in capturing detailed local structural information. The DGCNN method (Wang et al. 2019) (Fig. 3(b)) improved this only by incorporating the difference between feature vectors of points (directed edges in Fig. 3(b)) into the MLP learning process. However, these two methods still have limited capability for local geometric relationship perception and cannot fully mitigate the impact of the point cloud's sparsity and incompleteness.

The k-NN search may identify neighbors with inaccurate relevance to the target points due to sparse sampling, as illustrated in Fig. 4(a), where only the two closest points (orange)

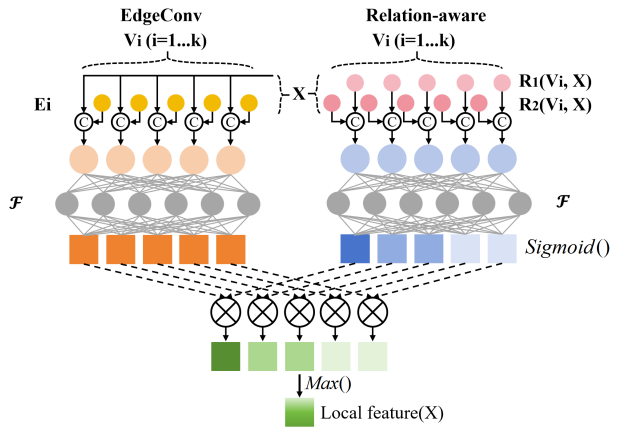

Figure 5: The framework of local geometric relationship perception.

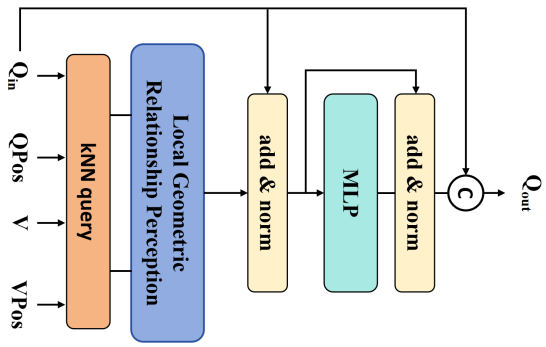

Figure 6: Scale-tailored local feature extractor.

show significant correlation with the target points (red), and more distant points (green) may provide little or even misleading information. To solve this problem, we propose a scheme (Fig. 3(c)) that introduces an adaptative contribution weight for each nearest neighbor based on its correlation with the target point. Specifically, we quantify the interrelationships between a target point and its neighboring points in terms of geometry in 3D space and feature vectors in high-dimensional space simultaneously. The quantified values are then utilized in a learning-based approach to assess each neighbor's contribution to the target point.

As previously discussed and substantiated by the examples presented in Fig. 4(a), the spatial distance between two points significantly impacts their geometric correlation. Accordingly, we use the Manhattan distance between the target point Q and its nearest neighbor point V_i $(i = 1, \ldots, k)$ in different directions as the spatial relationship metric $R1$. This is formulated as follows:

$$R_1(Q, V_i) = \left(\left|x_Q - x_{V_i}\right|, \left|y_Q - y_{V_i}\right|, \left|z_Q - z_{V_i}\right|\right). \quad (1)$$

To measure the relationship between points in high-dimensional feature space, a simple method is to directly extend the $R1$ metric to a high dimension. Nevertheless, it does not produce satisfactory results. Therefore, we design a more intricate yet effective relationship metric as shown in Fig 4(b), denoted as $R2$. Initially, we select a small subset (M points) of the nearest neighbors and calculate their average relative relationship (average of directed edges) with the target point. This average vector represents the *primary* change trend of the target point within a small region. We then subtract this average vector from the directed edge between the target point and each of its neighbors and take the absolute value to serve as the relationship metric between the two points. It's noteworthy that the $R1$ is derived from the direct relation between points, while $R2$ is derived from the relative relation (the edge between two points). The formulation of $R2$ is as follows:

$$R_2\left(Q, V_i\right) = \left| \frac{1}{M} \sum_{j=1}^{M} \operatorname{vec}\left(Q, V_j\right) - \operatorname{vec}\left(Q, V_i\right) \right|, \quad (2)$$

where vec represents the vector of differences (directed edges) between two points.

Fig. 5 illustrates the technical implementation of local geometric relationship perception. Specifically, the left part utilizes the process of EdgeConv for local feature extraction. The right part is our proposed relationship perception. We concatenate the relationship metrics $R1$ and $R2$ and use an MLP to output contribution weights in a learning manner. Then, the generated weights are element-wise multiplied by the extracted features. Finally, robust and geometry-related features are generated through a max function. We provide a more detailed analysis of local geometric perception with respect to the relationship metrics $R1$ and $R2$ in the Supplementary Material(Zhong et al. 2024).

Progressive Feature Extractor

Many point cloud completion approaches utilize the multi-layer DGCNN (Yu et al. 2021) consisting of EdgeConv and farthest point sampling (FPS) at each layer. This configuration allows for simultaneous down-sampling and feature extraction of the input incomplete point cloud. However, this approach fails to capture global point relationships. Recently, PointAttn (Wang et al. 2024) introduced a purely attention-based feature extraction method that establishes long-distance dependencies using a global self-attention mechanism, which is computationally expensive. To leverage the strengths of both approaches and incorporate our relation-based local geometric feature extraction, we develop a progressive feature extraction process that moves from coarse to fine granularity in the feature level while slowly reducing the point's number (from dense to sparse configurations), efficiently balancing computational cost and extraction accuracy.

As depicted in the lower left corner of Fig. 2, our method is structured into two phases. In the initial phase, the input incomplete point cloud is subjected to a single-layer DGCNN for preliminary sampling and feature extraction, where we conducted a relatively dense sampling. Due to the single-layer operation, the features obtained at this stage are relatively coarse. The second phase involves a progressive refinement of feature extraction, with two alternating rounds of global relationship querying via self-attention modules and local relationship querying via the ST-LFE module. The

design of ST-LFE enables both feature dimensionality expansion and point scale pruning, with its internal structure detailed in Fig. 6. Consequently, the dense and coarse features in the first phase are refined in the second phase, transforming into sparser but more intricately expressed sampling point features.

Point Proxy Correction Module

In previous studies, the feature of point proxies within the Transformer's decoder was facilitated through a cross-attention mechanism with the hidden features output by the encoder, enabling an implicit information exchange. While the point proxies predicted by the encoder are with high-dimensional features, their inherent sparsity falls short of capturing the dense distribution characteristics of the original 3D point cloud, thereby leading to a partial information loss. To enrich and enhance the representation of point agent features, we propose a correction module as shown in Fig. 2. This module reuses a dense input point cloud, employing a processing flow akin to the feature extractor we have developed in Section . Specifically, within a low-dimensional space, this module explicitly conveys dense point cloud information to the generated point proxies through self-attention and a local feature extractor. Consequently, point proxies are endowed with comprehensive information about the dense point cloud, more accurately reflecting the distribution characteristics of the original input data.

Network Optimization

For the loss function, we adopt a setting similar to AdaPoinTr. The first training objective is to minimize the Chamfer distance between the predicted point cloud (consisting of a sparse point proxy set called "$\mathcal{C}$" and a final dense prediction point set called "$\mathcal{P}$") and the actual point cloud (called "$\mathcal{G}$"), thereby ensuring a closer approximation to the true point cloud configuration. It is formulated as:

$$J_0 = \frac{1}{n_{\mathcal{C}}} \sum_{c \in \mathcal{C}} \min_{g \in \mathcal{G}} \|c - g\| + \frac{1}{n_{\mathcal{G}}} \sum_{g \in \mathcal{G}} \min_{c \in \mathcal{C}} \|g - c\|, \quad (3)$$

$$J_1 = \frac{1}{n_{\mathcal{P}}} \sum_{p \in \mathcal{P}} \min_{g \in \mathcal{G}} \|p - g\| + \frac{1}{n_{\mathcal{G}}} \sum_{g \in \mathcal{G}} \min_{p \in \mathcal{P}} \|g - p\|. \quad (4)$$

The second training objective is a denoising task designed to enhance the model's robustness to random noise. Given a noisy query $\hat{\mathcal{Q}}_i$ and corresponding noisy center $\hat{c}_i^{gt} = n_i + c_i^{gt}$, the model aims to reconstruct the detailed local shape centered at c_i^{gt}, despite the presence of noise n_i. We denote the true local shape centered at c_i^{gt} as $\mathcal{G}_{c_i}^{gt}$, and the local shape predicted from $\hat{\mathcal{Q}}_i$ as $\hat{\mathcal{P}}_i$, the auxiliary loss function can be expressed as follows:

$$J_{\text{denoise}} = \frac{1}{\left|\hat{\mathcal{P}}_i\right|} \sum_{c \in \hat{\mathcal{P}}_i} \min_{g \in \mathcal{G}_{c_i}^{gt}} \|c-g\| + \frac{1}{\left|\mathcal{G}_{c_i}^{gt}\right|} \sum_{g \in \mathcal{G}_{c_i}^{gt}} \min_{c \in \mathcal{P}_i} \|g-c\|. \quad (5)$$

To this end, the final objective for our network is $J_{PC} = J_0 + J_1 + \lambda J_{\text{denoise}}$.

Experiments

Datasets and Implementation Details

We train PointCFormer on several datasets, including PCN (Yuan et al. 2018), ShapeNet-55 (Yu et al. 2021), ShapeNet-34/Unseen-21 (Yu et al. 2021) and KITTI. To ensure a fair comparison, we adhere to the standard protocols for training and testing on each dataset. PointCFormer is implemented with PyTorch and trained on two NVIDIA 4090 GPUs. The internal transformer encoder consists of six cascaded standard self-attention blocks, and the point cloud generation and transformer decoder scheme are similar to the Adapointr framework. Our network is trained using the AdamW optimizer with a base learning rate set to 0.0001. We use L1/L2 Chamfer distance and F-Score (Yu et al. 2023) as evaluation metrics for the PCN, ShapeNet-55, and ShapeNet34/Unseen21. On KITTI, we report the fidelity distance (FD) and minimum matching distance (MMD).

Comparisons with State-of-the-Art Methods

We compare our method with several advanced methods, including FoldingNet (Yang et al. 2018), PCN (Yuan et al. 2018), GRNet (Xie et al. 2020), PoinTr (Yu et al. 2021), SnowFlakeNet (Xiang et al. 2021), SeedFormer (Zhou et al. 2022), AnchorFormer (Chen et al. 2023), PointAttN (Wang et al. 2024), and AdaPointr (Yu et al. 2023).

Evaluation on PCN. The PCN dataset comprises 28,974 shapes for training and 1,200 shapes for testing, distributed across eight categories. It is currently the most frequently utilized benchmark dataset for evaluating the performance of point cloud completion methods. As shown in Table 1, our method leads in almost all metrics. Specifically, our method improves the average CD-ℓ_1 by 0.43 compared to the recently proposed PointAttN and shows remarkable improvements across different categories. Compared to the 3DMambaCom, our average CD-ℓ_1 improves by 0.5, and we see a 0.31 increase in the F-Score@1%.

Fig. 7 illustrates visual comparisons of different methods on the PCN dataset. PointCFormer yields results that are closer to the ground truth. Specifically, the input point clouds shown in the top two rows exhibit unique features of their original shapes (i.e., cars and airplanes). While most adopted methods can reconstruct general 3D structures, our PointCFormer demonstrates advantages in capturing complex local details (e.g., the wings of an airplane and the wheels of a car). Moreover, when faced with the two sets of challenging point cloud inputs shown below, our PointCFormer achieves the most accurate recovery of the overall shape, outperforming other methods in both fidelity and accuracy.

Evaluation on ShapeNet-55. Next, we evaluate PointCFormer on the ShapeNet-55 dataset, which encompasses more categories. Table 2 presents the performance of various methods on incomplete point cloud data with three different missing ratios (CD-ℓ_2-S, CD-ℓ_2-M, and CD-ℓ_2-H) in terms of L2 Chamfer distance (CD-ℓ_2) and F-Score@1%. PointCFormer evidently outperforms other existing methods in CD-ℓ_2-S, CD-ℓ_2-M, CD-ℓ_2-H, and average CD-ℓ_2. Despite scoring slightly lower than the top-performing method

Methods	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat	Avg CD-ℓ_1	F-Score@1%
FoldingNet(cvpr2019)	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31	0.322
PCN(3dv2018)	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64	0.695
GRNet(eccv2020)	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83	0.708
PoinTr(iccv2021)	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38	0.745
Snowflake(2021iccv)	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21	-
SeedFormer(eccv2022)	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85	6.74	-
AnchorFormer(cvpr2023)	3.97	9.59	8.53	8.46	6.39	9.13	6.73	6.16	7.37	-
PointAttN(aaai2024)	3.88	17.923	9.01	7.28	5.97	-	-	-	6.84	-
3DMambCom(arxiv2024)	3.86	9.11	7.72	7.41	5.73	9.04	6.29	6.09	6.91	0.824
AdaPoinTr(tpami2023)	3.68	8.82	7.47	6.85	5.47	8.35	**5.80**	5.76	6.53	0.845
PointCFormer	**3.53**	**8.73**	**7.32**	**6.68**	**5.12**	**8.34**	5.86	**5.74**	**6.41**	**0.855**

Table 1: Performance comparison on the PCN dataset. We use the CD-ℓ_1(multiplied by 1000) and F-Score@1% to compare with other methods.

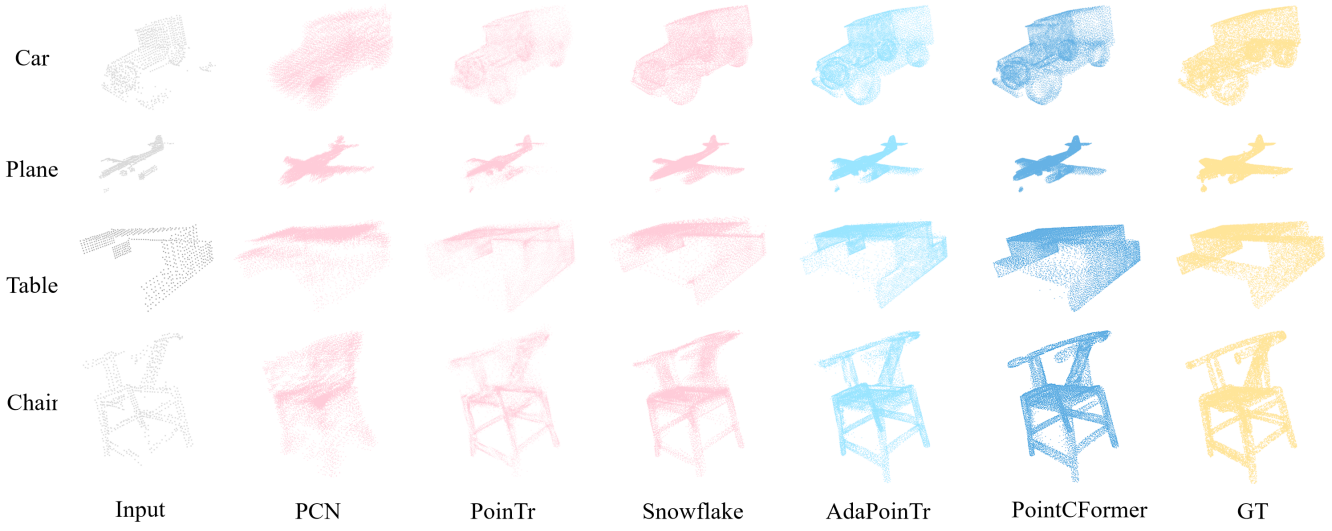

Figure 7: Visual examples of point cloud completion results on the PCN dataset using different methods.

Method	CD-ℓ_2-S	CD-ℓ_2-M	CD-ℓ_2-H	CD-ℓ_2-Avg	F-Score@1%
FoldingNet	2.67	2.66	4.05	3.12	0.082
PCN	1.94	1.96	4.08	2.66	0.133
PoinTr	0.67	1.05	2.02	1.25	0.446
Snowflake	0.81	1.17	2.20	1.40	0.343
AnchorFormer	1.14	1.12	1.91	1.39	0.327
PointAttN	0.47	0.66	1.17	0.77	-
3DMambCom	0.61	0.77	1.20	0.86	0.341
AdaPoinTr	0.49	0.69	1.24	0.81	**0.503**
PointCFormer	**0.42**	**0.64**	**1.15**	**0.73**	0.499

Table 2: Point cloud completion results on ShapeNet-55.

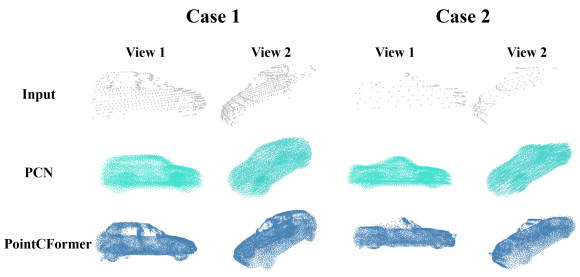

Figure 8: Qualitative results on KITTI: dual-view representation of the same point cloud for enhanced visualization of car shapes in each scenario.

(AdaPoinTr) in terms of F-Score@1%, we achieve state-of-the-art performance with a CD-ℓ_2 of 0.73. These results underscore the effective completion capabilities of PointCFormer, even when applied to such a diverse dataset.

Evaluation on ShapeNet-34/Unseen-21 We conduct experiments on ShapeNet-34/unseen21 to evaluate the generalization ability of PointCFormer. The results presented in Table 3 indicate that despite a close F1 score with the runner-up under various ratio settings, PointCFormer consistently achieves the best performance on the CD metric across both the 34 visible categories and the 21 unseen categories. This reflects its excellent generalization ability.

	34 seen categories					21 unseen categories				
	CD-S	CD-M	CD-H	CD-ℓ_2-Avg	F-Score@1%	CD-S	CD-M	CD-H	CD-ℓ_2-Avg	F-Score@1%
FoldingNet	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
PCN	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
PoinTr	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
Snowflake	0.60	0.86	1.50	0.99	0.422	0.88	1.46	2.92	1.75	0.388
AnchorFormer	0.85	1.09	1.77	1.23	0.328	1.09	1.58	2.88	1.85	0.289
PointAttN	0.51	0.70	1.23	0.81	-	0.76	1.15	2.23	1.38	-
3DMambCom	0.66	0.84	1.39	0.96	0.324	0.86	1.34	2.99	1.73	0.281
AdaPoinTr	0.48	0.63	1.07	0.73	**0.469**	0.61	0.96	2.11	1.23	0.416
PointCFormer	**0.41**	**0.55**	**1.03**	**0.66**	0.459	**0.53**	**0.88**	**1.97**	**1.12**	**0.420**

Table 3: Performance comparison on ShapeNet-34: results for three difficulty levels across 34 seen and 21 unseen categories. CD-S, CD-M, and CD-H represent CD-ℓ_2 (multiplied by 1000) results under simple, medium, and hard settings, respectively.

*1000	SnowflakeNet	SeedFormer	PointAttN	AdaPoinTr	PointCFormer
Fidelity↓	0.110	0.151	0.672	0.237	**0.001**
MMD↓	0.907	0.516	0.504	0.392	**0.353**

Table 4: Quantative comparison on KITTI in terms of MMD and Fidelity.

Evaluation on KITTI KITTI dataset comprises incomplete point clouds of real vehicles captured by LiDAR scans of world scenes. Following the protocol in (Xie et al. 2020), we fine-tune our model initially trained on ShapeNet-Cars (Yuan et al. 2018) and assess its performance on the KITTI dataset. As reported in Table 4, PointCFormer consistently outperforms other models across the fidelity and MMD metrics. This illustrates the superior capability of PointCFormer in capturing the 3D shape characteristics of vehicles. Furthermore, Fig. 8 shows two examples of point cloud completion across two views. PointCFormer exhibits superior overall quality with finer local details in granular patterns, reflecting the advantages of leveraging the novel feature extraction paradigm we have introduced for enhancing the network's ability to parse 3D structures.

Ablation Study

We conducted a comprehensive ablation study on the PCN dataset to demonstrate the effectiveness of each module we designed within PointCFormer. The results are summarized in Table 5. Here, "LGRP" refers to a progressive feature extractor that contains only two ST-LFE modules, "GSP" refers to a progressive feature extractor that uses only two self-attention layers (Applying LGRP and GSP is the whole PFE.), and "CM" stands for the correction module. The baseline "I" is a combination of the partial framework from AdaPoinTr with a conventional encoder. We observe a significant performance boost when we replace the traditional local feature extraction process with "LGRP". The model's performance was further enhanced by incorporating the global shape perception process (GSP). Lastly, the model's best performance was obtained after adding the correction module. It is evident from the results that the local geometric relationship perception contributes the most to the model's improvement, showcasing its great potential for other point cloud tasks.

	LGRP	GSP	CM	CD-ℓ_1↓	F-Score@1%↑
I				6.92	0.810
II	✓			6.47	0.840
III		✓		6.85	0.822
IV			✓	6.87	0.819
V		✓	✓	6.81	0.828
VI	✓	✓		6.44	0.853
VII	✓		✓	6.48	0.844
VIII	✓	✓	✓	**6.41**	**0.855**

Table 5: Ablation study of several modules proposed in PointCFormer on the PCN Dataset.

Conclusions

In this paper, we explore preserving extensive global structural features while extracting detailed local features in point cloud completion tasks. We introduce PointCFormer, a novel point cloud completion model that builds upon the Transformer framework. A key component is the local relation perception module, which measures the contribution of each point within a local region based on the relations in 3D space and high-dimensional feature space. Another key component is the progressive feature extractor combined with the attention mechanism to strengthen the capture of global structures. Additionally, the latent space correction module ensures that the generated point proxies align more accurately with the distribution of the original input point cloud data. Crucially, we incorporate a relation-based local perception mechanism into both the feature extractor and the correction module, significantly enhancing their performance. Comprehensive comparative and ablation studies across various challenging benchmarks demonstrate that PointCFormer achieves superior completion performance. In fact, our relation-based feature extraction method comprehensively considers the global and local information of point clouds. We would like to extend our method to other point cloud-related tasks in the future.

Acknowledgments

This work is partially funded by the National Natural Science Foundation of China (12494550 and 12494553),

the Beijing Natural Science Foundation (Z240002), and the Guangdong Science and Technology Program (2023B1515120026).

References

Bi, Z.; and Wang, L. 2010. Advances in 3D data acquisition and processing for industrial applications. *Robotics and Computer-Integrated Manufacturing*, 26(5): 403–413.

Chen, C.; and Yang, B. 2016. Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119: 90–107.

Chen, Z.; Long, F.; Qiu, Z.; Yao, T.; Zhou, W.; Luo, J.; and Mei, T. 2023. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13581–13590.

Dai, A.; Ruizhongtai Qi, C.; and Nießner, M. 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5868–5877.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fei, B.; Yang, W.; Ma, L.; and Chen, W.-M. 2023. DcTr: Noise-robust point cloud completion by dual-channel transformer with cross-attention. *Pattern Recognition*, 133: 109051.

Gu, A.; and Dao, T. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv preprint arXiv:2312.00752*.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7: 187–199.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

Han, X.; Li, Z.; Huang, H.; Kalogerakis, E.; and Yu, Y. 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE international conference on computer vision*, 85–93.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, Y.; Yang, W.; and Fei, B. 2024. 3dmambacomplete: Exploring structured state space model for point cloud completion. *arXiv preprint arXiv:2404.07106*.

Mao, A.; Yan, B.; Ma, Z.; and He, Y. 2024. Denoising Point Clouds in Latent Space via Graph Convolution and Invertible Neural Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5768–5777.

Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; and Xu, C. 2021. Voxel transformer for 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3164–3173.

Misra, I.; Girdhar, R.; and Joulin, A. 2021. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2906–2917.

Pan, X.; Xia, Z.; Song, S.; Li, L. E.; and Huang, G. 2021. 3d object detection with pointformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7463–7472.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Stutz, D.; and Geiger, A. 2018. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1955–1964.

Tang, J.; Gong, Z.; Yi, R.; Xie, Y.; and Ma, L. 2022. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1726–1735.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, 10347–10357. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Cui, Y.; Guo, D.; Li, J.; Liu, Q.; and Shen, C. 2024. PointAttN: You Only Need Attention for Point Cloud Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6): 5472–5480.

Wang, P.-S. 2023. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4): 1–11.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12.

Wang, Y.; Tan, D. J.; Navab, N.; and Tombari, F. 2022. Learning local displacements for point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1568–1577.

Wen, X.; Xiang, P.; Han, Z.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Liu, Y.-S. 2022. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 852–867.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xiang, P.; Wen, X.; Liu, Y.-S.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Han, Z. 2021. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 5499–5509.

Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; and Sun, W. 2020. Grnet: Gridding residual network for dense point cloud completion. In *European conference on computer vision*, 365–381. Springer.

Yan, X.; Yan, H.; Wang, J.; Du, H.; Wu, Z.; Xie, D.; Pu, S.; and Lu, L. 2022. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, 676–693. Springer.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 206–215.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12498–12507.

Yu, X.; Rao, Y.; Wang, Z.; Lu, J.; and Zhou, J. 2023. AdaPoinTr: Diverse Point Cloud Completion With Adaptive Geometry-Aware Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 14114–14130.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, 728–737. IEEE.

Zhang, W.; Zhou, H.; Dong, Z.; Liu, J.; Yan, Q.; and Xiao, C. 2022. Point cloud completion via skeleton-detail transformer. *IEEE Transactions on Visualization and Computer Graphics*, 29(10): 4229–4242.

Zhong, Y.; Quan, W.; ming Yan, D.; Jiang, J.; and Wei, Y. 2024. PointCFormer: a Relation-based Progressive Feature Extraction Network for Point Cloud Completion. arXiv:2412.08421.

Zhou, H.; Cao, Y.; Chu, W.; Zhu, J.; Lu, T.; Tai, Y.; and Wang, C. 2022. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision*, 416–432. Springer.