Dig into Detailed Structures: Key Context Encoding and Semantic-based Decoding for Point Cloud Completion

Somebody

ABSTRACT

Recovering the complete shape of a 3D object from limited viewpoints plays an important role in 3D vision. Encouraged by the effectiveness of feature extraction using deep neural networks, recent point cloud completion methods prefer an encoding-decoding architecture for generating the global structure and local geometry from a set of input point proxies. In this paper, we introduce an innovative completion method aimed at uncovering structural details from input point clouds and maximizing their utility. Specifically, we improve both Encoding and Decoding for this task: (1) Key Context Fusion Encoding extracts and aggregates homologous key context by adaptively increasing the sampling bias towards salient structure and special contour points that are more representative of object structure information. (2) Semantic-based Decoding introduces a semantic EdgeConv module to prompt next Transformer decoder, which effectively learns and generates local geometry with semantic correlations from non-nearest neighbors. The experiments are evaluated on several 3D point cloud and 2.5D depth image datasets. Both qualitative and quantitative evaluations demonstrate that our method outperforms previous state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Shape inference**.

KEYWORDS

Point Cloud Completion; Generative Model; 3D-Context

ACM Reference Format:

Somebody. 2018. Dig into Detailed Structures: Key Context Encoding and Semantic-based Decoding for Point Cloud Completion. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (MM '24).* ACM, New York, NY, USA, 10 pages. https://doi.org/XX

1 INTRODUCTION

With the popularity and development of 3D scanning equipment, 3D point clouds have wide applications in computer vision, such as autonomous driving [4], robotics [15], and industrial detection [19]. However, the original point clouds captured by 3D scanners or depth cameras in real-world are often incomplete due to self-occlusion, limited viewpoints, and low resolution. Therefore, recovering the complete and dense 3D shape from partial point clouds is important in many downstream 3D tasks [18, 20].

MM '24, October 28-November 1, 2024, Melbourne, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XX

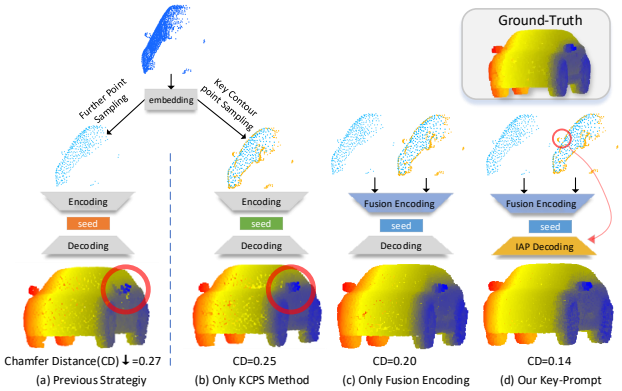

Figure 1: Illustration of our main idea. We first compare the point cloud generated by a traditional down-sampling-based method (a) with that produced by our 'Key Contour Point Sampling' enhanced feature extraction (b). We further investigated the application of a fused encoding (c) and the semantic Interest Area Prompt (IAP) decoding (d), which also represents our proposed Key-Prompt model which provides more details and reduces the Chamfer distance metrics.

Since deep neural networks have been successful in extracting features from 3D coordinates of point clouds, recent point cloud completion methods [14, 39] tend to use an encoding-decoding architecture for the coarse-to-fine generation. The encoding phase extracts features from the partial point cloud utilizing methods such as PointNet [22] or DGCNN [30]. A set of sparse points is generated via input features to sketch the global structure (called seeds), while the decoding phase generates a complete point cloud by refining the seed's local geometry. However, during the above processing, existing methods have two problems that still need to be solved: 1) The initially dense input point cloud needs to be aggregated several times during encoding, a process that often loses abundant geometric details. (2) Features are extracted from the incomplete input point cloud, posing challenges in directly generating the geometry of the correlated missing parts.

In the encoding phase, the necessary pre-processing downsampling strategy tends to abstract the spatial structure by merging a dense given point cloud into a component-level coarse point. Here, Farthest Point Sampling (FPS) [8] is a widely used method, which has an even sampling rate at each part, as illustrated in Fig. 1(a). To enhance the representativity of these point centers, methods vary the down-sample resolution or use different neighborhood scales for richer contexts. However, they neglected the fact that points that are located on contours or on complex structures often bring great help to the structural representation of the point cloud.

In the decoding phase, geometries of the missing parts are often recovered by constructing a relationship between sparse seed and input. The existing modeling of dependencies between seeds and

input points relies solely on neighboring geometric relationships, making it hard to learn long-distance correlation input information for seeds. Recent methods[17, 42] directly transformed the local features of the input part after self-attention coding into the features of missing part. AdaPoinTr[37] uses the geometry-aware module with both self-attention and cross-attention mechanisms to let seeds learn structural knowledge and detailed information. However, they pay more attention to the position and geometry embedding, and their usual correlations are built in the neighboring areas. In this case, those sparse seeds, especially of missing parts are hard to focus similarity local representation from incomplete input.

To solve the above problems, we propose a novel point cloud completion method with improved encoding-decoding architectures, as shown in Fig. 1. Firstly, a new adaptive down-sampling method is employed to maximize the preservation of key edge contours. Compared with FPS, the detailed geometries are well represented with the same ratio of down-sampled points (Fig. 1(b)). Then, two kinds of homologous down-sampling point cloud features are fused by a kind of self Vector Attention crosswise in our encoding phase to discover more input key context (Fig. 1(c)). After the encoding phase generates a set of seeds, we notice that semantic similarity is an essential cue for building local correlations between seeds and the partial point cloud, but it is always ignored. For example, similar semantic structures sometimes occur in symmetrical but distant locations. Thus, we design a semantic cross-attention decoding. This module initially acquires semantic structure for each seed proxy and then selectively acquires knowledge on the local geometry of missing parts via point-to-area attention learning as a prompt for the next transformer decoder. With richer feature extraction encoding and enhanced semantic guidance decoding, our method performs excellently for point cloud completion (Fig. 1(d)).

The main contributions of this work are summarized as follows:

- We propose a KCPS method that preserves richer geometry details via dynamically enforcing larger sampling biases toward key contour. Integrating KCPS and FPS extractor strategies, we transform given point clouds into input local proxies without sacrificing its contour descriptions.
- We propose a semantic-based IAP decoding method that introduces semantics cues into the dependency inference among given and inferred points, thus significantly improving point cloud completion with richer structure priors.
- Our method achieves state-of-the-art performance on various benchmarks, including 3D point cloud datasets such as PCN, Project shapenet55-34 and KITTI, as well as Redwood RGBD datasets.

2 RELATED WORK

2.1 Point Cloud Representation and Completion

Most of the early work in point cloud completion utilized 3D voxel grids [7, 13, 16, 25, 29] as an intermediate representation for each voxel block, which was then processed by methods such as 3D convolution. However, the voxelization will lose details of the point cloud, and the computational cost increases heavily in voxel resolution. The advent of PointNet revolutionized this landscape [22], enabling direct feature extraction from unstructured and disordered point clouds using deep neural networks. Unlike PointNet, which operates on individual points, PointNet++ [23] leverages k-nearest neighbor search and maximum pooling operations to aggregate neighborhood context information, progressively abstracting local regions along hierarchical scales by repeatedly employing Farthest Point Sampling [8]. DGCNN [30] introduces the EdgeConv module, which maps the representation of edges into the feature space and dynamically aggregates point cloud neighborhoods multiple times. This dynamic graph structure enhances the network's ability to learn point cloud representations with semantics. PCN [39] is the first learning-based method for point cloud completion. After PCN, many methods [14, 27, 32, 42] adopt a similar coarse-to-fine architecture to enhance the real details of point cloud generation. SD-Net [3] introduces an end-to-end disentangled completion structure comprising two subnetworks for input refinement and missing part prediction, facilitating detail cross-recovery at various scales. Similarly, CRAPCN [24] incorporates a local attention mechanism during the encoding and decoding phase to fuse contextual information from multi-resolution point clouds. However, many of these methods incorporate various point contexts through different scales of down-sampling resolution or different numbers of neighbor aggregations, which often leads to overlooking detailed structures and crucial contour points. This oversight can have a significant impact on the representation and generation of whole structures.

2.2 Transformer in Point Cloud Completion

With the remarkable success of Transformer [28] in natural language processing, numerous applications have surged into the realm of computer vision. In the realm of point cloud completion, the unstructured set nature of point clouds prompted the introduction of Transformers to more effectively address the limitations of previous architectures in exploring associations within collections. Guo et al. [12] introduced PCT, which enhances the self-attention mechanism and incorporates local point cloud neighborhood information for embedding, making the Transformer more suitable for point cloud's vector attention. Therefore, many models based on the Transformer architecture accomplish detail completion by constructing relationships between seeds and inputs. Point Transformer combined the Vector Attention [41] with a U-Net style encoder-decoder framework. Seedformer [42] introduces a feature Up-Transformer module and the novel representation concept of patch seed, efficiently integrating spatial relations between neighboring points transformed into seeds. However, the self-attention decoding in Seedformer makes it hard to extract meaningful information from sparse seed points. Consequently, many Transformer-based models [2, 5, 17, 35] facilitate more detailed completion by establishing correlations between seeds and input data. PoinTr [37] and its variant [38] reformulate the point cloud completion problem as a set-to-set transformation and adopt a novel Transformer architecture to realize efficient completion. During the decoding phase, they utilize the geometry-aware module to establish a link between seed and input data, leveraging attention mechanism to learn implicit correlation from the input. However, due to the inherent disparities between input and seed representing missing parts, the set-to-set attention mechanism and current decoding architecture do not solve the problem of establishing associations between the two sets for interactive learning.

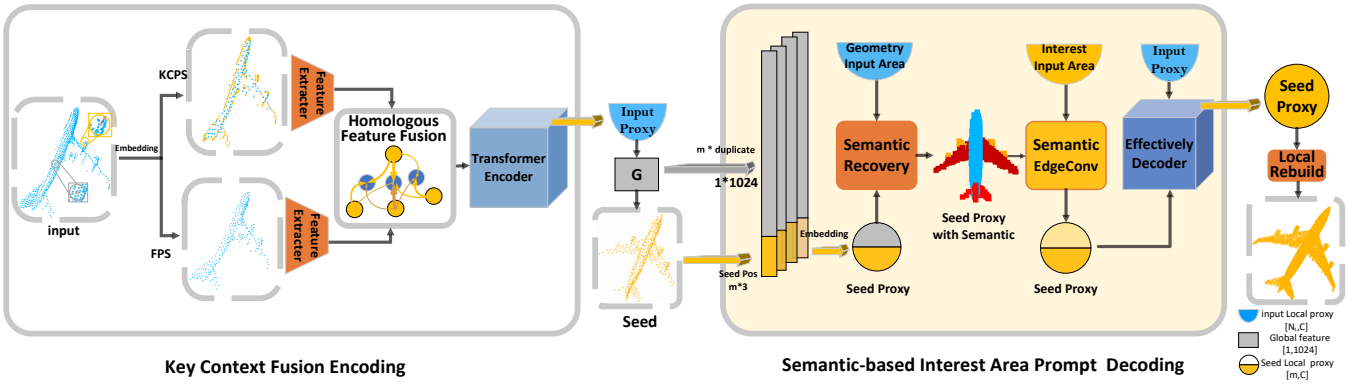

Figure 2: The overall architecture of Key-Prompt. We use double extractors with FPS and KCPS to learn and fuse these homologous features with different focuses. While generating seeds, the architecture preserves semantic information of the feature extractors and the component association in the global structure as much as possible. The embedded semantic Edgeconv will prompt interest input areas for the seed proxy, facilitating richer eventual refinement.

3 METHOD

3.1 Problem Formulation

There are multiple task representations of point cloud completion. Following the design of [37, 38], we define the point cloud complete task as a set-to-set transform task, which is well suited to the attention mechanism. Consider an input F-dimensional point cloud with N points, denoted by $\mathbf{P}_{in} = \{p_{in}^1, \ldots, p_{in}^N\} \subseteq \mathbb{R}^{N\times F}$, in the simplest setting of $F = 3$, each point contains 3D coordinates $p_{in}^i = (x_i, y_i, z_i)$. The final point cloud of the complete M points is represented as $\mathbf{Y} = \{y_1, \ldots, y_M\} \subseteq \mathbb{R}^{M\times F}$.

The overall network architecture of Key-Prompt is shown in Fig. 2, and we next introduce a few of the critical processing variables follow our architecture. To take advantage of the rich local details, Key Contour Point Sampling (KCPS) is used to construct a new point cloud $\mathbf{P'} \subseteq \mathbb{R}^{N_1\times F}$ in parallel with the FPS point cloud $\mathbf{P} \subseteq \mathbb{R}^{N_1\times F}$ for subsequent feature extraction and fusion. Here both the subset $\mathbf{P'}$ and $\mathbf{P}$ are down-sampled from input point cloud $\mathbf{P}_{in}$. Local points are converted into feature vectors representing a localized region, called input point proxy $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{N_1}\} \subseteq \mathbb{R}^{N_1\times C}$, which means that C-dimensional point cloud with N_1 down-sampled points. We transform the input point proxies into seeds $\mathbf{P}_q$ and seed proxies Q via a global feature $\mathbf{G}$ for semantic preserving. The Interest Area Prompt decoding structure selects valid information from the interest input to enrich the representation of the seed proxy. Finally, we recover the local structure near each seed based on its proxy to generate the final point cloud $\mathbf{Y}$.

3.2 Key Context Fusion Encoding

3.2.1 Key Contour Point Sampling. Currently, down-sampling is often performed to aggregate fine-grained point clouds in the local vicinity during feature extraction. Traditional Farthest Point Sampling (FPS) [8] tends to result in the loss of points on delicate structures and contours, as shown in Fig. 3. Compared to flat surfaces, these points usually carry more structure information and local variations in computing neighbor edge vectors, which are particularly crucial for understanding an object's details and global structure.

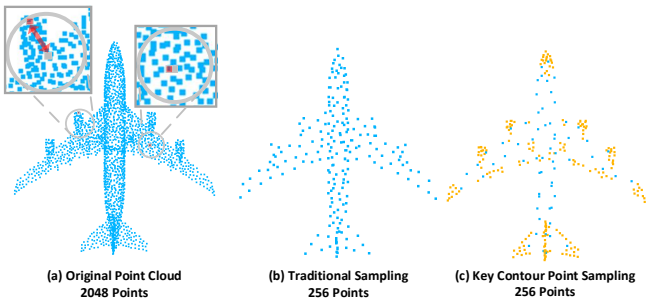

Figure 3: Our Sample Method. Figure (a) shows the difference between points on the contour and inside the structure calculating the offset; (b) and (c) are the results we obtained using traditional FPS method and our KCPS method.

Such points are defined as 'Key Contour Points'. We have adapted and refined Ahmed's method [1], which defines contour points based on the offset from their neighborhood centers, allowing us to identify critical points more effectively. The neighborhood of a point p_i is identified through a k-nearest neighbor search, represented as $\mathrm{V}_i = \{n_1, n_2, \ldots . n_k\}$. Subsequently, the center coordinates C_i of these neighboring points are calculated by incorporating each point n_i within the vicinity:

$$C_i = \frac{1}{k}\sum_{j=1}^{k} n_j \tag{1}$$

To mitigate differences in density between regions and ensure scale invariance, the point dispersion distance D_i of the region is calculated as follows:

$$D_i\,(\,\mathrm{V}_i) = \max_{n\in \mathrm{V}_i} \|p_i - n_i\| \tag{2}$$

Then, a degree of importance I_i is defined to quantify the offset from the center while adaptive by vicinity densities as follows:

$$I_i = \frac{\|\,p_i - C_i\,\|}{D_i} \tag{3}$$

The input point cloud is sorted and sampled by the importance level I_i to dynamically distinguish their differences in computing neighboring point edges. The top $\frac{q}{2}$ points are sampled as keypoints, for the remaining points, we use the FPS, stitching the two point clouds together to obtain the reconstructed point cloud $\mathbf{P}'$.

In contrast to Ahmed's method [1], we use the maximum value in calculating D_i to achieve greater robustness, especially with non-uniform point distributions. We also find that calculating too few neighborhood points may only select points to outline shapes, while too many can obscure critical structural areas. A moderate number of neighbors is often best for identifying salient contour keypoints. Fig. 3 illustrates that our method results in more accurately capturing intricate structures with the same sampling ratio.

3.2.2 Double Extractor and Context Fusion. Aimed at capturing both the broad structure and intricate details, we propose a dual-extractor framework depicted in Fig. 2. Starting with the embedded input point cloud, two sampling techniques are applied for the next feature extraction. We utilize FPS to obtain coordinates $\mathbf{P}$, which helps in understanding the overall shape, while KCPS acquires coordinates $\mathbf{P}'$, focusing on structure details. Then, points and their corresponding embedded features are fused to ensure that the detail-focused insights are integrated with the overall view provided.

In each feature extractor, we follow DGCNN [30], first build a local graph structure based on local neighborhoods for the point cloud, noting the edge vector connecting the neighbor p_j of the p_i as $p_j - p_i$. We connect the edge vector with p_i and feed them into the neural network. The steps are as follows:

$$\mathcal{F}_{ij} = \mathrm{ReLU}\left(Liner\left(\mathrm{x}_i, \mathrm{x}_j - \mathrm{x}_i\right)\right) \tag{4}$$

$$\mathcal{F}_i = Max(F_{p_{ij}}), \forall j : p_j \in \kappa\left(p_i\right) \tag{5}$$

where $\mathrm{x}_i, \mathrm{x}_j$ represents the point's embedded feature, we gather the features using max-pooling as Max for whose coordinates p_j locates within the k-neighborhood of p_i, represented as $\kappa(p_i)$. For ease of writing, we combine these two EdgeConv steps and write them in a formula as follows:

$$\mathcal{F}_i = \mathrm{EdgeConv}(x_i, x_j), \forall j : p_j \in \kappa\left(p_i\right) \tag{6}$$

It is worth emphasizing here that similar geometrical structures will yield similar semantics. EdgeConv is actually a dynamic graph attention mechanism based on subtraction relations. Our dual-extraction strategy, using multiple EdgeConv layers, conducts a secondary sampling based on each distinct sampling method, thereby dynamically updating the graph structure independently. This process not only refreshes the information in the salient structure but also dynamically disseminates details and semantic information throughout the point cloud.

We innovatively use Vector Attention [41] to fuse KCPS and FPS-derived features and coordinates. Vector Attention was initially used for self-attention in one point cloud, but our two point clouds are homologous which makes it easy to extend it to cross-point cloud feature fusion. Instead of learning each point in the set globally, we refine each point proxy $\mathcal{F}_i$, based on its local neighborhood spatial coordinates p'_j and corresponding proxy $\mathcal{F}_j$, enhancing the structural comprehension of the whole point cloud. Compared to the dot product, the β-subtracted relational function are better reflection. Relative position coding denoted as δ is additionally added into fusion:

$$\delta = \mathrm{Liner}(p_i - p'_j), \forall j : p'_j \in \kappa\left(p_i\right) \tag{7}$$

$$\mathcal{F}_i = \rho\left(\gamma\left(\beta\left(\phi\left(\mathcal{F}_i\right), \psi\left(\mathcal{F}_j\right)\right) + \delta\right) \odot \alpha\left(\mathcal{F}_j + \delta\right)\right) \tag{8}$$

where ϕ, ψ, and α are linear projection functions, and the mapping function γ is an MLP with two linear layers and one ReLU nonlinearity. ρ is a normalization function such as softmax. Therefore, we apply the neighbor area attention mechanism to fuse each query point cloud p_i with p'_j, while incorporating more locally informative features F_j into final input local proxy $\mathcal{F}_i$.

3.3 Semantic-based Decoding

In this section, we innovatively introduce an Interest Area Prompt Decoding shown in Fig. 2 that significantly helps seed to achieve selective learning of relevant information in the input. We define the seed proxy as $\mathcal{Q}$, incomplete input feature as $\mathcal{K}, \mathcal{V}$, and formulate the Transformer encoder $\mathcal{M}_E$ and decoder $\mathcal{M}_D$ layers processes as follows:

$$\mathcal{V} = \mathcal{M}_E(\mathcal{F}), \quad \mathcal{H} = \mathcal{M}_D(\mathcal{Q}, \mathcal{V}) \tag{9}$$

The current decoding method [5, 38] only builds correlations based on geometric distances, ignoring the missing parts' seeds to focus on the most relevant areas, as shown in Fig. 4 (b). Our method recovers the seed semantic information shown in Fig. 4 (c) while successfully focusing on the input empennage and wing located on the symmetric side using the similarity of the semantic features

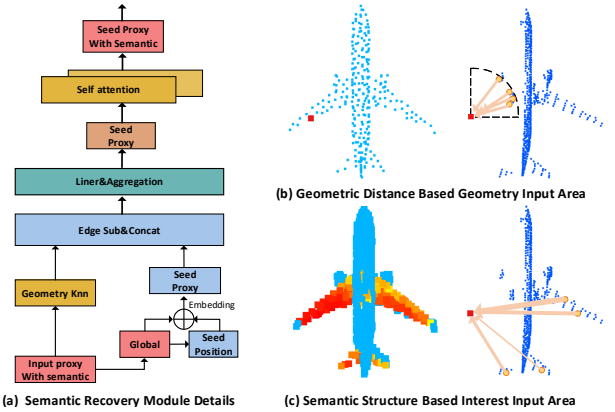

Figure 4: We show details of semantic recovery module and the difference of input Area of interest for geometric-based and semantic-based EdgeConv.

We detail the process of recovering potential semantic information, utilizing geometric and self-attention mechanisms, in Fig. 4 (a). Each input local proxy are initially through Transformer Encoder and embedding, subsequently aggregating them into global features denoted as $\mathbf{G}$. Following this step, we proceed to generate a sparse seed $\mathbf{P}_q$ along with its corresponding seed proxy, represented as $\mathcal{Q}$:

$$\mathcal{Q}_i = [\mathbf{G}, p_q^i] = [Max(\mathcal{V}), p_q^i] \tag{10}$$

The input proxy $\mathcal{F}$ and global features $\mathbf{G}$ could encompass semantic structures, thanks to our dynamic EdgeConv extractor's capability to infuse features with semantic significance. This principle is rooted in observation that regions with geometrical similarities

also tend to exhibit similar local neighborhood graph structures during feature extraction. Semantic similarity is evident in geometric structures found within one component or between similar components, such as empennage and the wings of an airplane. Then, using geometric distance-based EdgeConv, we quickly capture the semantic features of neighboring inputs, aiding in the retrieval of semantic information from the input:

$$\phi\left(Q_i\right) = \text{EdgeConv}(Q_i, \mathcal{K}_j - Q_i), \forall j : p_k^j \in \kappa\left(p_q^i\right), \quad (11)$$

Further, we use self-attention to assist in self-similar structural associations originating from global features $\mathbf{G}$, with the ability to capture long-range dependencies:

$$Q = \mathcal{M}_E(Q) \quad (12)$$

In this way, points in the same or related structure can learn similar semantics from each other. We use the distance in the feature space to measure the degree of semantic similarity, as shown Fig. 4 (c). So we could use seed local proxy Q with semantic information to construct new graph structure relations with an interest area of input proxy $\mathcal{K}$ in the feature space:

$$\tau\left(Q_i\right) = \text{EdgeConv}(Q_i, \mathcal{K}_l - Q_i), \forall l : K_l \in \kappa\left(Q_i\right), \quad (13)$$

This brings the semantically corresponding partial features closer to missing parts, which amounts to a directed cue to the decoder of the subsequent set to set. So we call it Interest Area Prompt (IAP). We then perform a more focal and effective Transformer decoder of Q and $\mathcal{V}$, as shown in Eq. 9.

Afterwards, we refine the sparse seed into a dense and complete point cloud by converting the seed proxy Q into the displacement deviation $\mathcal{H}$ of the neighborhood point through a simple MLP module. The final point cloud is represented as:

$$\mathbf{Y}_i^k = \mathcal{H}_i^k + p_q^i, k = \frac{M}{m} \quad (14)$$

where m and k are the number of seed points and each seed localized points, resulting in a point cloud $\mathbf{Y} \subseteq \mathbb{R}^{M\times 3}$ of M points.

3.4 Loss Function

Fan et al. [9] proposed the Chamfer Distance (CD) as an evaluation metric for the 3D reconstruction task:

$$d_{\text{CD}}\left(P_1, P_2\right) = \frac{1}{P_1}\sum \min_{y\in P_2} \|x-y\|_2^2 + \frac{1}{P_2}\sum \min_{x\in P_1} \|y-x\|_2^2 \quad (15)$$

The CD distance is used to measure the nearest-square distance between the predicted point cloud P_1 and the ground-truth point cloud P_2, which is computed for each of the two point cloud sets and averaged.

To ensure the sparse seed $\mathbf{p}_q$ adequately represents the ground truth shape, the ground truth is down-sampled to match the seed point count denoted by $\mathbf{Y}_{gt}^1$. This step allows for the calculation of the loss $\mathcal{L}_s$ by comparing $\mathbf{p}_q$ with Y_{gt}^1. For assessing the quality of the finally refined generated result, the comparison with the ground truth generates the final result loss, $\mathcal{L}_f$. In addition, following the denoising strategy from AdaPoinTr [38], noise is introduced to the seed points during training. The goal is to predict the distribution of these noise to ground truth values in order to make the Transformer decoder more robust to the initial seed. The CD between the noise-adjusted generation and the corresponding amount of ground truth, denoted by $\mathbf{Y}_{gt}^2$, is then evaluated to quantify the noise elimination effectiveness, resulting in the loss $\mathcal{L}_d$:

$$\mathcal{L}_s = d_{\text{CD}}(p_q, \mathbf{Y}_{gt}^1), \mathcal{L}_f = d_{\text{CD}}(p_q, \mathbf{Y}_{gt}), \mathcal{L}_d = d_{\text{CD}}(p_q, \mathbf{Y}_{gt}^2) \quad (16)$$

The training loss function $\mathcal{L}$ is finally defined as:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_f + \lambda\mathcal{L}_d \quad (17)$$

4 EXPERIMENTS

In this section, we conducted experiments on point cloud completion, including real-world scenes, with validation on more challenging 2.5D depth images. We compared our approach to baseline methods on benchmark datasets like PCN and the more challenging Project-ShapeNet-55/34. We also pre-trained our model on PCN and tested it on real-world datasets like KITTI and Redwood RGBD, outperforming existing approaches both qualitatively and quantitatively.

4.1 Experimental Settings

4.1.1 Implementation Details. We implement Key-Prompt with Pytorch [21]. All models are trained using two NVIDIA Tesla V100 gpus. The ablation study and complexity analysis are conducted under the same experimental conditions. For the training phase, we utilize AdamW for optimizing our model, setting the initial learning rate to 10^{-4}, and 400 epochs will be conducted with a learning rate decay set to 0.9. Moreover, considering that the local neighborhood of the KCPS point cloud is denser and richer, we incorporate a larger number of neighbors (24) to extract meaningful key context for feature extraction while using 16 neighbors for point clouds of FPS.

4.1.2 Evaluation Metrics. We adhere to prior research practices and utilize the average Chamfer distance(CD) as our primary evaluation metric, which is widely referenced in the field. As mentioned in section 3.4, a higher average CD indicates a greater disparity between the two point cloud sets. Specifically, the CD significantly increases when the generated point cloud contains extra or missing parts compared to the ground truth. Additionally, we employ the F-score metric, following the methodology introduced by Tatarchenko et al. [26], to evaluate the proportion of similar components. This metric gauges similarity by assessing whether a point from one set falls within a certain threshold of another set.

4.2 Point Cloud Completion

4.2.1 Evaluation on PCN Dataset. **Data.** The PCN dataset [39] is one of the most commonly used datasets in the field of point cloud completion. It is a subset of ShapeNet including 8 categories with a total of 30974 CAD models. To emulate real-world sensor data, incomplete point clouds were generated by back-projecting 2.5D depth images from 8 viewpoints. These created incomplete point clouds consist of 2048 points, while the ground truth contains 16,384 points sampled from the complete point cloud. We adopt the identical experimental setting as PCN [39] to ensure a fair comparison with existing models.

Results. We provide Chamfer Distance and F-Score@1% results for the L1 norm (×1000) in Tab. 1, comparing recent models. Our method consistently outperforms prior state-of-the-art methods

Table 1: Completion results on PCN dataset in terms of per-point L1 Chamfer Distance ×1000 (lower is better) and F1-score.

	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	CD-ℓ_1	F-Score@1%
FoldingNet [36]	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31	0.322
TopNet [27]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15	0.503
AtlasNet [11]	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61	10.85	0.616
PCN [39]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64	0.695
GRNet [34]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83	0.708
PoinTr [37]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38	0.745
NSFA [40]	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48	8.06	-
PMP-Net++ [31]	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56	-
SnowflakeNet [33]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21	-
SeedFormer [42]	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85	6.74	-
ProxyFormer [17]	4.01	9.01	7.88	7.11	5.35	8.77	6.03	5.98	6.77	-
AdaPoinTr [38]	3.68	8.82	7.47	6.85	5.47	8.35	5.80	5.76	6.53	0.845
Key-Prompt	**3.43**	**8.70**	**7.30**	**6.56**	**5.05**	**8.20**	**5.76**	**5.55**	**6.32**	**0.857**

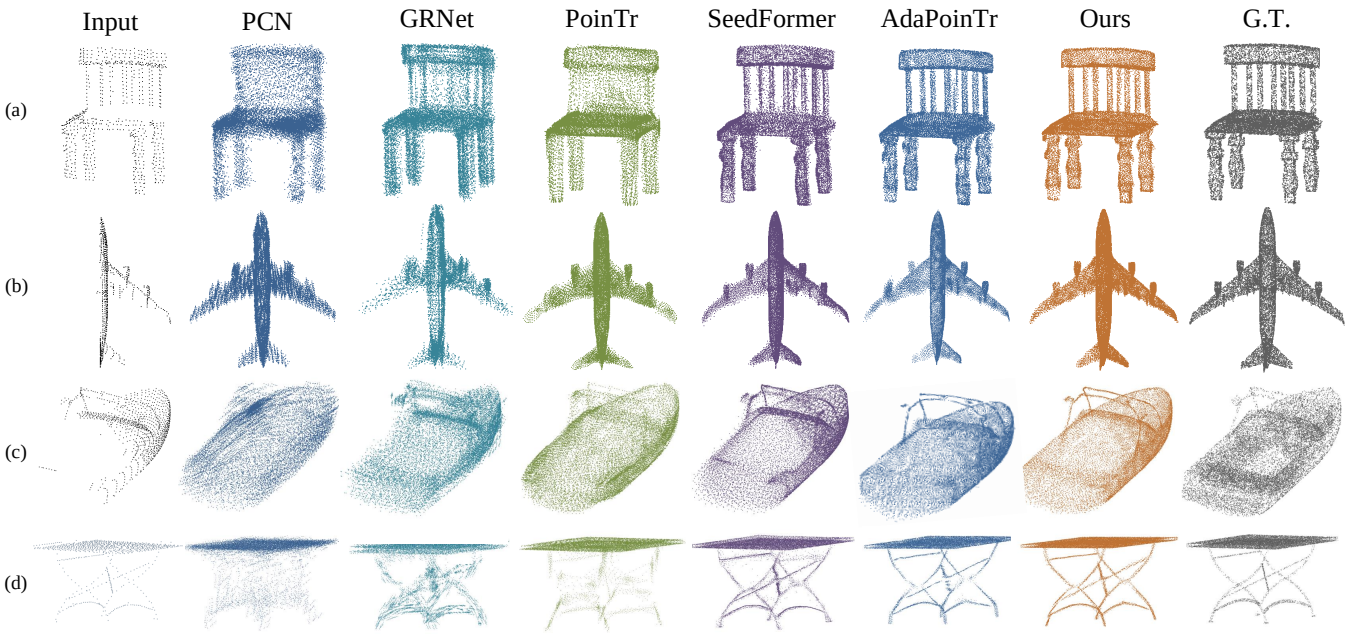

Figure 5: Qualitative comparisons on the PCN dataset are shown. Each method completes the object from the incomplete point cloud shown at the far left of each row. Our method produces completion that most closely matches the ground truth.

across all categories, with notably superior average metrics. Additionally, in Fig. 5, we visually compare shapes from chair, plane, boat, and table categories with SeedFormer [42], AdaPoinTr [38], and other methods[34, 37, 39]. These comparisons highlight Key-Prompt's ability to generate highly specific and realistic results with richer detailed information, especially for missing parts. For instance, as seen in the chair backrest example in Fig. 5(a), our method delineates a much clearer structure. *More visualization results and shows of semantic interest areas are available in the Appendix.*

4.2.2 Evaluation on Project ShapeNet-55/34. **Data.** The Project-ShapeNet-55 dataset encompasses a wider array of object categories and introduces different methods of simulating incomplete point clouds. It incorporates the entirety of ShapeNet's 55 categories, where both training and testing are conducted on this comprehensive set. The unique aspect of Project-ShapeNet-55 is its method of generating partial inputs, specifically through a noisy back projection technique. This process involves rendering each sample from 16 randomly selected viewpoints to create depth images. Subsequently, these depth images are utilized to simulate incomplete point clouds by employing a noisy back projection method, effectively replicating scenarios encountered in the real world. The Project-ShapeNet-34 dataset is designed to test the generalization capabilities of various models. Utilizing 34 categories for training purposes sets up a unique challenge for models to accurately predict an additional unseen 21 categories during the training phase while the remaining design is consistent with Project-ShapeNet-55.

Table 2: Completion results on Project-ShapeNet-55 dataset

	Many-Shot categories					Few-Shot categories						
	Table	Chair	Airplane	Car	Sofa	Microwaves	Bag	Pillow	Keyboard	Rocket	CD-ℓ_1	F-Score
PCN [39]	14.79	15.33	9.07	12.85	17.12	39.54	18.64	20.04	13.69	10.98	16.64	0.403
TopNet [27]	14.40	16.29	9.85	13.61	16.93	26.64	18.69	19.57	11.05	10.45	16.35	0.337
GRNet [34]	12.01	12.57	8.30	12.13	14.36	18.35	14.67	15.15	9.71	8.58	12.81	0.491
SnowflakeNet [33]	10.49	11.07	6.35	11.20	12.59	16.92	12.86	14.02	8.12	7.49	11.34	0.594
PoinTr [37]	9.97	10.43	6.02	10.58	12.11	17.06	12.15	12.57	7.61	6.86	10.68	0.615
AdaPoinTr [38]	8.81	9.12	5.18	9.77	10.89	17.43	10.93	11.82	6.79	5.58	9.58	0.701
Key-Prompt	**8.69**	**8.96**	**5.13**	**9.60**	**10.77**	**13.64**	**10.06**	**11.21**	**6.64**	**5.44**	**9.35**	**0.714**

Table 3: Completion results on Project-ShapeNet-34 dataset

	21 unseen categories				
	Pillow	Skateboard	Earphone	CD-ℓ_1	F-Score
PCN [39]	23.45	17.27	24.82	21.44	0.307
TopNet [27]	17.55	12.59	19.34	15.98	0.358
GRNet [34]	18.64	10.60	15.00	15.03	0.439
SnowflakeNet [33]	15.35	9.58	15.19	12.82	0.551
PoinTr [37]	14.82	8.98	14.23	12.43	0.555
AdaPoinTr [38]	13.72	8.34	12.30	11.37	0.642
Key-Prompt	**13.32**	**7.43**	**11.39**	**10.97**	**0.654**

For the evaluation strategy, we align with the methodology described by Yu et al. [37], where the ground truth point clouds are represented by 8,192 points. The generated incomplete point clouds used as input are configured to contain 2,048 points.

Results on Projected-ShapeNet-55 The introduction of noise into the input point cloud contributes to the dataset's complexity and realism, thus presenting a heightened challenge for completion. We provide an analysis of our models and existing methodologies, outlining the class-specific Chamfer Distance (CD) and overall CD for each approach. Specifically focusing on ten categories, we present detailed results. Tab. 2 demonstrates that Key-Prompt outperforms other approaches across these categories and in terms of overall CD. The Many-Shot categories on the left have more than 2500 train samples, while the Few-Shot categories contain less than 80 train samples. Our results greatly outperform AdaPoinTr [38], suggesting that our method is able to mine more valid information from fewer training sets.

Results on Projected-ShapeNet-34 In Tab. 3, we present the performance of our model and compare it with existing models. The data clearly indicate that our model outperforms others in all unseen categories. Importantly, our model not only could improve results in familiar categories but also significantly better identify unseen categories, such as Pillow, Skateboard, and Earphone. This performance highlights our model's strong generalization ability, showing its effectiveness in classifying new object types that were not included in the training dataset.

4.3 Real Scene Point Cloud Completion

4.3.1 Evaluation on KITTI. **Data.** To evaluate our model's performance with real-world data, we conducted experiments on the KITTI [10] dataset sourced from LIDAR scans. The KITTI dataset, widely recognized in autonomous driving research, presents a challenge due to the sparsity of LIDAR-derived data. Thus, generating complete and dense point clouds is crucial for downstream tasks, such as 3D target detection. Specifically, we extracted and localized the point clouds of car objects as input by employing a 3D bounding box within each frame. Since this dataset does not have a complete point cloud as ground truth, we follow the GRNet [34] using Fidelity Distance (FD) and Minimal Matching Distance (MMD) as evaluation metrics.

Results. Following the GRNet's [34] setup, we pre-trained our model on the Car category of the PCN [39] dataset to complement the incomplete cars in KITTI. As shown in Tab. 4, our model outperforms several baseline models. The car point clouds generated by our model accurately depict windows and tires while capturing intricate details, such as reflectors, as evidenced in Fig. 6.

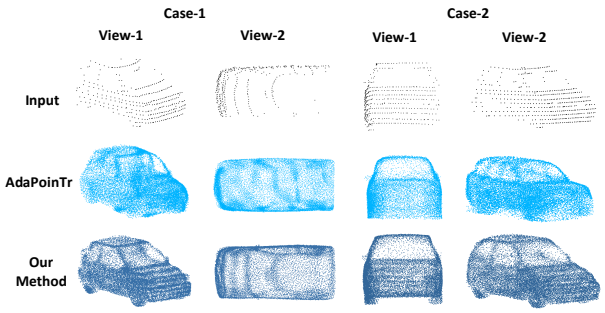

Figure 6: Qualitative results on the KITTI dataset. We show two different views of each object while our method can recover a car with more accurate contour and details.

4.3.2 Depth Image Completion to 3D. Exploring the efficacy of models pre-trained on the PCN [39] dataset, we engaged with real object scans from the Redwood 3D Scans [6] dataset, focusing on localized shapes of chairs, sofas, and cars derived from depth images captured with an RGB-D camera. Unlike LiDAR data, point clouds from real-depth images typically suffer from lower accuracy, more noise, and increased occlusion complexities. Given the scarcity of complete meshes in the Redwood 3D Scans dataset, our evaluation was qualitative, assessing the model's ability to recover original object shapes and details with high fidelity. The experimental results in Fig. 7, illustrate that our predictions demonstrate superior recovery of the original object shape and local details.

Table 4: KITTI Dataset results. The comparison between the following models is based on the FD and MMD metrics.

CDl2(x1000)	AtlasNet [11]	PCN [39]	FoldingNet [36]	TopNet [27]	PFNet [14]	GRNet [34]	SeedFormer [42]	AdaPoinTr [38]	**Key-Prompt**
Fidelity	1.759	2.235	7.467	5.354	1.137	0.816	0.151	0.237	**0.136**
MMD	2.108	1.366	0.537	0.636	0.792	0.872	0.516	0.392	**0.380**

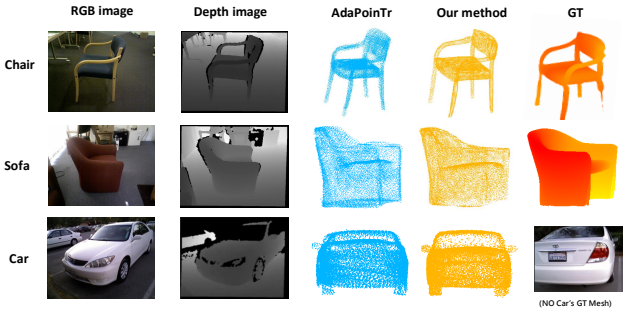

Figure 7: Results of the chair, sofa, and car in Redwood. We show the input point cloud and ground truth, as well as the prediction results from AdaPoinTr and our method.

4.4 Ablation Study

In this section, we demonstrate the effectiveness of several module designs in Key-prompt. We replace our method with some traditional modules for ablation experiments, and all our networks are trained and tested on PCN datasets with the same settings.

Sampling Methods. We take different sampling methods for feature extraction. We perform separate (A) Farthest Point Sampling and (A1) Key Contour Point Sampling, and we find that using KCPS alone gives only a slight boost but could better recover some detailed structures. Therefore, we utilize both sampling methods as follows: (B) Certain key contour points are directly incorporated into the output of the farthest point sampling as spatial fusion, and they persist as input for the subsequent Feature Extractor.

Fusion Encoding. Our other strategy is to use two feature extractors to take the two sampling methods separately and fuse the output features. We have alternative fusion schemes for feature fusion:(C) Feature fusion using traditional attentional mechanism fusion and (D) Vector Attention (VA) based feature fusion method. Experiments have shown that Vector Attention mechanism gives better results. We found that the method of VA fusion originally used for one point cloud can be better applied to our type of fusion of homologous point clouds and features.

Interest Area Prompted Decoding. Traditional point cloud completion models based on transformer cross-decoder use the basic set to set Transformer in the decoding phase. We designed the IAP module to change this decoding structure, which allows for models with fewer parameters. Results in Tab. 5 show clear improvement of our IAP design (6.32 vs 6.53).

4.5 Accuracy-Complexity Trade-Offs

Our dual-feature extractor architecture, operating in parallel, adds little to the computational complexity. Thanks to the IAP module, which significantly enhances the efficiency of the Transformer decoder architecture by sharpening the focus of attention. We are able to decrease the number of decoder layers without substantial accuracy loss. Consequently, we introduce a lightweight variant of Key-Prompt, equipped with only 5 Transformer decoders compared to AdaPoinTr's 8 decoders, achieving over a 20% reduction in FLOPS and a nearly 1.5 times faster computation, all while incurring a mere 1% dip in accuracy. We compared the model with existing GRNet [34], Seedformer [42], AdaPoinTr [38]. Our lightweight version of Key-Prompt achieves a better result than AdaPoinTr [38] with fewer FLOPs, and our method can process more than 40 point clouds in one second (using a batch size of 1). In our testing environment, Key-Prompt stands out as the fastest among recent models.

Table 5: Ablation Study. The table proves the validity of our three module designs respectively.

Model	Sample	Fusion	IAP	CD	F1-score
A	FPS	-	-	6.95	0.805
A1	KCPS	-	-	6.93	0.810
B	FPS+KCPS	Spatial	-	6.72	0.819
C	FPS+KCPS	Attn	-	6.64	0.821
D	FPS+KCPS	VA	-	6.55	0.842
Key-Prompt	FPS+KCPS	VA	✓	**6.32**	**0.857**

Table 6: Complexity Analysis. Comparisons of GFLOPS and Throughput of our models with other models.

Models	FLOPs	Throughput	CD
GRNet [34]	40.4 G	32.4 pc/s	6.74
Seedformer [42]	53.8 G	12.7 pc/s	6.74
AdaPoinTr [38]	15.1 G	31.8 pc/s	6.53
Key-Prompt	18.2 G	28.1 pc/s	**6.32**
Key-Prompt*	**14.2 G**	**40.2 pc/s**	6.39

Model marked with * represent our light version of Key-Prompt.

5 CONCLUSION

In this study, we introduce an innovative architecture named Key-Prompt, focusing on the detailed structures often overlooked by previous methodologies. During the encoding phase, we implement a novel sampling technique known as KCPS. Based on this method, we perform a deep fusion of point cloud features by integrating different sampling methods. Moreover, we designed a unique Semantic-based Interest Area Prompt decoding phase to enable coarse point cloud seed to learn key local details from the input accurately. Our method has achieved state-of-the-art performance on several challenging benchmark datasets through comprehensive experimental validation. Notably, the exceptional generalization ability of our model is underscored in tasks involving few-shot learning and predictions for unseen categories. Additionally, our model exhibits enhanced suitability for real-world applications, highlighting its practical significance and potential for broad adoption.

REFERENCES

[1] Syeda Mariam Ahmed, Yan Zhi Tan, Chee Meng Chew, Abdullah Al Mamun, and Fook Seng Wong. 2018. Edge and Corner Detection for Unorganized 3D Point Clouds with Application to Robotic Welding. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 7350–7355. https://doi.org/10.1109/IROS.2018.8593910

[2] Rui Cao, Kaiyi Zhang, Yang Chen, Ximing Yang, and Cheng Jin. 2022. Point Cloud Completion via Multi-Scale Edge Convolution and Attention. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*. Association for Computing Machinery, New York, NY, USA, 6183–6192. https://doi.org/10.1145/3503161.3548360

[3] Junxian Chen, Ying Liu, Yiqi Liang, Dandan Long, Xiaolin He, and Ruihui Li. 2023. SD-Net: Spatially-Disentangled Point Cloud Completion Network. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. Association for Computing Machinery, New York, NY, USA, 1283–1293. https://doi.org/10.1145/3581783.3611716

[4] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl K. Wellington. 2020. 3D Point Cloud Processing and Learning for Autonomous Driving: Impacting Map Creation, Localization, and Perception. *IEEE Signal Processing Magazine* 38 (2020), 68–86. https://api.semanticscholar.org/CorpusID:229703747

[5] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. 2023. AnchorFormer: Point Cloud Completion from Discriminative Nodes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 13581–13590. https://doi.org/10.1109/CVPR52729.2023.01305

[6] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. 2016. A Large Dataset of Object Scans. https://doi.org/10.48550/arXiv.1602.02481 arXiv:1602.02481 [cs]

[7] Angela Dai, Charles R Qi, and Matthias Nießner. 2017. Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 6545–6554. https://doi.org/10.1109/CVPR.2017.693

[8] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. 1997. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing* 6, 9 (1997), 1305–1315. https://doi.org/10.1109/83.623193

[9] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. 2016. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2463–2471. https://api.semanticscholar.org/CorpusID:6746759

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

[11] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2018. A Papier-Mache Approach to Learning 3D Surface Generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 216–224. https://api.semanticscholar.org/CorpusID:3656527

[12] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. 2021. PCT: Point Cloud Transformer. *Computational Visual Media* 7, 2 (June 2021), 187–199. https://doi.org/10.1007/s41095-021-0229-5

[13] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. 2017. High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 85–93. https://doi.org/10.1109/ICCV.2017.19

[14] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. 2020. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 7659–7667. https://doi.org/10.1109/CVPR42600.2020.00768

[15] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. 2021. Coarse-to-Fine Q-attention: Efficient Learning for Visual Robotic Manipulation via Discretisation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 13729–13738. https://api.semanticscholar.org/CorpusID:235606348

[16] Truc Le and Ye Duan. 2018. PointGrid: A Deep Network for 3D Shape Understanding. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Salt Lake City, UT, 9204–9214. https://doi.org/10.1109/CVPR.2018.00959

[17] Shanshan Li, Pan Gao, Xiaoyang Tan, and Mingqiang Wei. 2023. ProxyFormer: Proxy Alignment Assisted Point Cloud Completion with Missing Part Sensitive Transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 9466–9475. https://doi.org/10.1109/CVPR52729.2023.00913

[18] Ming Liang, Binh Yang, Shenlong Wang, and Raquel Urtasun. 2018. Deep Continuous Fusion for Multi-sensor 3D Object Detection. In *European Conference on Computer Vision*. https://api.semanticscholar.org/CorpusID:52211898

[19] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. 2017. Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction. *ArXiv* abs/1706.07036 (2017). https://api.semanticscholar.org/CorpusID:12973135

[20] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Nießner. 2020. RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 4606–4616. https://api.semanticscholar.org/CorpusID:227228047

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv* abs/1912.01703 (2019). https://api.semanticscholar.org/CorpusID:202786778

[22] 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 (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 77–85. https://doi.org/10.1109/CVPR.2017.16

[23] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 5105–5114.

[24] Yi Rong, Haoran Zhou, Lixin Yuan, Cheng Mei, Jiahao Wang, and Tong Lu. 2024. CRA-PCN: Point Cloud Completion with Intra- and Inter-level Cross-Resolution Transformers. https://doi.org/10.48550/arXiv.2401.01552 arXiv:2401.01552 [cs]

[25] David Stutz and Andreas Geiger. 2018. Learning 3D Shape Completion from Laser Scan Data with Weak Supervision. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1955–1964. https://doi.org/10.1109/CVPR.2018.00209

[26] Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. 2019. What Do Single-View 3D Reconstruction Networks Learn? *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 3400–3409. https://api.semanticscholar.org/CorpusID:115147778

[27] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. 2019. TopNet: Structural Point Cloud Decoder. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 383–392. https://doi.org/10.1109/CVPR.2019.00047

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[29] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017. O-CNN: octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph.* 36, 4, Article 72 (jul 2017), 11 pages. https://doi.org/10.1145/3072959.3073608

[30] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38, 5, Article 146 (oct 2019), 12 pages. https://doi.org/10.1145/3326362

[31] Xin Wen, Peng Xiang, Yaru Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. 2022. PMP-Net++: Point Cloud Completion by Transformer-Enhanced Multi-Step Point Moving Paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2022), 852–867. https://api.semanticscholar.org/CorpusID:247011998

[32] Yaqi Xia, Yan Xia, Wei Li, Rui Song, Kailang Cao, and Uwe Stilla. 2021. ASFM-Net: Asymmetrical Siamese Feature Matching Network for Point Completion. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21)*. Association for Computing Machinery, New York, NY, USA, 1938–1947. https://doi.org/10.1145/3474085.3475348

[33] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. 2021. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 5479–5489. https://api.semanticscholar.org/CorpusID:236965763

[34] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. 2020. GRNet: Gridding Residual Network for Dense Point Cloud Completion. *ArXiv* abs/2006.03761 (2020). https://api.semanticscholar.org/CorpusID:219531703

[35] Rui Xu, Le Hui, Yuehui Han, Jianjun Qian, and Jin Xie. 2023. Transformer-based Point Cloud Generation Network. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. Association for Computing Machinery, New York, NY, USA, 4169–4177. https://doi.org/10.1145/3581783.3612226

[36] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 206–215. https://doi.org/10.1109/CVPR.2018.00029

[37] Xumin Yu, 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). 12478–12487. https://doi.org/10.1109/ICCV48922.2021.01227

[38] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. 2023. AdaPoinTr: Diverse Point Cloud Completion With Adaptive Geometry-Aware Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 12 (2023), 14114–14130. https://doi.org/10.1109/TPAMI.2023.3309253

[39] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. 2018. PCN: Point Completion Network. In *2018 International Conference on 3D Vision (3DV)*. 728–737. https://doi.org/10.1109/3DV.2018.00088

[40] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. 2020. Detail Preserved Point Cloud Completion via Separated Feature Aggregation. *ArXiv* abs/2007.02374 (2020). https://api.semanticscholar.org/CorpusID:220364616

[41] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. 2021. Point Transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 16239–16248. https://doi.org/10.1109/ICCV48922.2021.01595

[42] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. 2022. SeedFormer: Patch Seeds Based Point Cloud Completion with Upsample Transformer. In *2022 European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, Heidelberg, 416–432. https://doi.org/10.1007/978-3-031-20062-5_24

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009