SVDFormer: Complementing Point Cloud via Self-view Augmentation and Self-structure Dual-generator

Zhe Zhu[1], Honghua Chen[1], Xing He[1], Weiming Wang[2†], Jing Qin[3], Mingqiang Wei[1†]
[1]Nanjing University of Aeronautics and Astronautics
[2]Hong Kong Metropolitan University
[3]The Hong Kong Polytechnic University
zhuzhe0619@nuaa.edu.com; chenhonghuacn@gmail.com; hexing@nuaa.edu.cn;
wmwang@hkmu.edu.hk; harry.qin@polyu.edu.hk; mqwei@nuaa.edu.cn

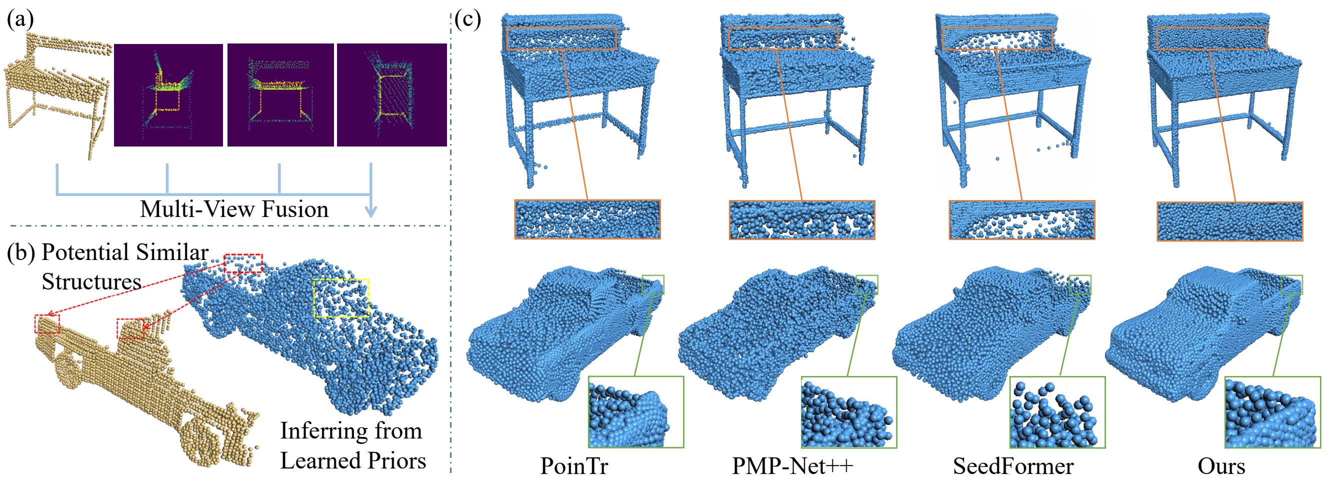

Figure 1. The characteristics of our method and visual comparison of point cloud completion results. (a) SVDFormer understands incomplete shapes from self-projected multiple views. (b) SVDFormer collaborates both geometric similarities (red boxes) and shape priors (yellow boxes) for shape refinement. (c) Qualitative comparison of our SVDFormer with PoinTr [43], PMP-Net++ [33], and SeedFormer [50].

Abstract

In this paper, we propose a novel network, SVDFormer, to tackle two specific challenges in point cloud completion: understanding faithful global shapes from incomplete point clouds and generating high-accuracy local structures. Current methods either perceive shape patterns using only 3D coordinates or import extra images with well-calibrated intrinsic parameters to guide the geometry estimation of the missing parts. However, these approaches do not always fully leverage the cross-modal self-structures available for accurate and high-quality point cloud completion. To this end, we first design a Self-view Fusion Network that leverages multiple-view depth image information to observe incomplete self-shape and generate a compact global shape. To reveal highly detailed structures, we then introduce a refinement module, called Self-structure Dual-generator, in which we incorporate learned shape priors and geometric self-similarities for producing new points. By perceiving the incompleteness of each point, the dual-path design disentangles refinement strategies conditioned on the structural type of each point. SVDFormer absorbs the wisdom of self-structures, avoiding any additional paired information such as color images with precisely calibrated camera intrinsic parameters. Comprehensive experiments indicate that our method achieves state-of-the-art performance on widely-used benchmarks. Code is available at `https://github.com/czvvd/SVDFormer`.

†Co-corresponding authors

1. Introduction

Point cloud completion plays an essential role in 3D vision applications and remains an active research topic in recent years. To tackle this task, a variety of learning-based techniques have been proposed, of which many demonstrate encouraging results [44, 14, 47, 43, 35, 38, 45, 23, 50, 46]. However, the sparsity and large structural incompleteness of captured point clouds still limit the ability of these methods to produce satisfactory results.

We observe that there are two primary challenges in this task. The first challenge is that crucial semantic parts may be absent, resulting in a vast solution space for point-based networks [44, 35, 43, 50] to identify plausible global shapes and locate missing regions. Some alternative methods attempt to address this issue by incorporating additional color images [48, 1, 51], but the paired images are hard to obtain, as well as the well-calibrated intrinsic parameters. The second one is how to infer detailed structures. Some recent methods [35, 38] utilize skip-connections between multiple refinement steps, allowing them to better leverage learned shape pattern priors to iteratively recover finer details. Some other methods prioritize preserving the original detail information by no-pooling encoding [50] or structural relational enhancement [18]. However, all above mentioned approaches typically employ a unified refinement strategy for all surface regions, which hinders the generation of geometric details for various missing regions. By observing and analyzing the partial inputs, we find that the missing surface regions can be classified into two types. The first type lacks similar structures in the input shape, and their reconstruction heavily relies on the learned shape prior. The second type is consistent with the local structures that are present in the partial input, and their recovery can be facilitated by appropriate geometric regularity [49]. For instance, LiDAR scans in KITTI [8] are highly sparse and contain limited information for generating fine details. Existing refinement strategies tend to produce and preserve implausible line-like shapes (see from Figure 7).

Based on the above observations, we propose a new neural network for point cloud completion called SVDFormer. Our method makes improvements by fully leveraging self-structure information in a coarse-to-fine paradigm.

First, similar to how a human would perceive and locate the missing areas of a physical object by observing it from different viewpoints, we aim to drive the neural network to absorb this knowledge by augmenting the data representation. To achieve this, we design a Self-View Fusion Network (SVFNet) that learns an effective descriptor, well depicting the global shape from both the point cloud data and depth maps captured from multiple viewpoints (see from Figure 1 (a)). To better exploit such kind of cross-modal information, we specifically introduce a feature fusion module to enhance the inter-view relations and improve the discriminative power of multi-view features.

Regarding the second challenge, our insight is to disentangle refinement strategies conditioned on the structural type of each point to reveal detailed geometric structures. Therefore, we design a Self-structure Dual-Generator (SDG) with a pair of parallel refinement units, called *Structure Analysis* and *Similarity Alignment*, respectively. The former unit analyzes the generated coarse point clouds by explicitly encoding local incompleteness, which enables it to match learned geometric patterns of training data to infer underlying shapes. The *Similarity Alignment* unit finds the features of similar structures for every point, thus making it easier to refine its local region by mimicking the geometry of input local structures. With the aid of this dual-path design, our method can generate reasonable results for different types of input shapes, including symmetrical synthetic models with various degrees of incompleteness and highly sparse real-world scans.

Extensive experiments demonstrate that SVDFormer achieves state-of-the-art performance on widely-used benchmarks. Our key contributions are listed below:

- We design a novel network called SVDFormer, which significantly improves point cloud completion in terms of global shape understanding and details recovery.
- We propose the novel Self-view Fusion Network (SVFNet) equipped with a feature fusion module to enhance the multi-view and cross-modal feature, which can output a plausible global shape.
- We introduce a Self-structure Dual-Generator (SDG) for refining the coarse completion. It enables our method to handle various kinds of incomplete shapes by jointly learning the local pattern priors and self-similarities of 3D shapes.

2. Related Work

2.1. Learning-based Shape Completion

Early learning-based methods [4, 9, 21, 25] often rely on voxel-based representations for 3D convolutional neural networks. However, these approaches are limited by their high computational cost and limited resolution. Alternatively, GRNet [37] and VE-PCN [27] use 3D grids as an intermediate representation for point-based completion.

In recent years, several methods are proposed to directly process points by end-to-end networks. One pioneering point-based work is PCN [44], which uses a shared multi-layer perceptron (MLP) to extract features and generates additional points using a folding operation [39] in a coarse-to-fine manner. Inspired by it, a lot of point-based methods [28, 16, 32, 35, 50, 43] have been proposed.

Later, to address the issue of limited information available in partial shapes using only point data, several works [48, 1, 51, 13] have explored the use of auxiliary in-

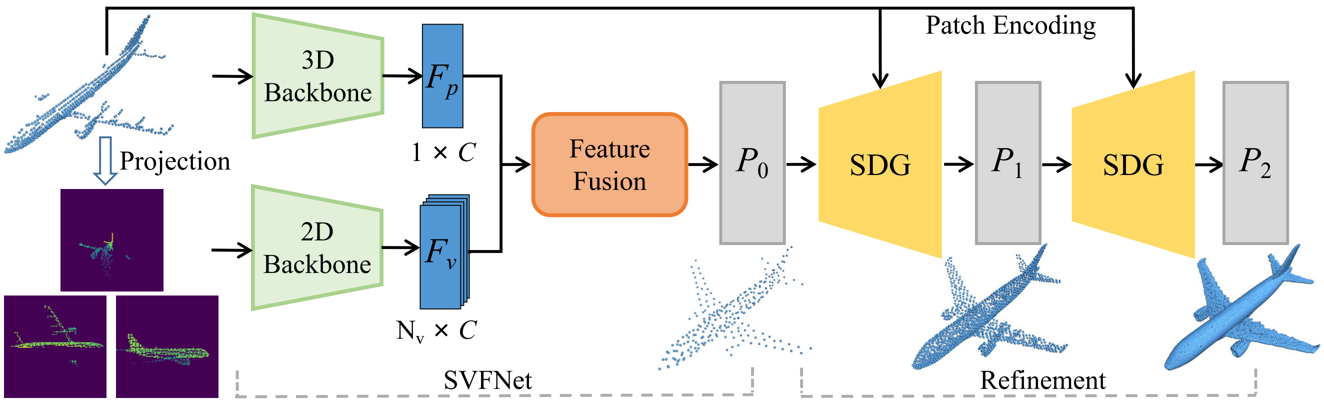

Figure 2. The architecture of SVDFormer. SVFNet first generates a global shape from the cross-modal input. The coarse completion is then upsampled and refined with two SDGs.

put to enhance performance. We call them cross-modal-based methods. These approaches involve the combination of rendered color images and partial point clouds, along with the corresponding camera parameters. Although these methods have shown promising results, they often require additional input that is difficult to obtain in practical settings. Different from these 3D data-driven methods, MVCN [12] operates completion solely in the 2D domain using a conditional GAN. However, it lacks the ability to supervise the results using ground truth with rich space information. Besides, some other methods [36, 1] seek to supervise point cloud completion in the 2D domain. The 2D projections of completed points are used to calculate loss by comparing them to the ground-truth depth. In contrast to these methods, we propose to utilize 2D input by observing self-structures to understand the overall shape. As a result, our method achieves a more comprehensive perception of the overall shape without requiring additional information or differentiable rendering during training.

Considering the high-quality details generation, a variety of strategies have been introduced by learning shape context and local spatial relationships. To achieve this goal, state-of-the-art methods design various refinement modules to learn better shape priors from the training data. SnowflakeNet [35] introduces Snowflake Point Deconvolution (SPD), which leverages skip-transformer to model the relation between parent points and child points. FB-Net [38] adopts the feedback mechanism during refinement and generates points in a recurrent manner. LAKe-Net [23] integrates its surface-skeleton representation into the refinement stage, which makes it easier to learn the missing topology part. Another type of method tends to preserve and exploit the local information in partial input. One direct approach is to predict the missing points by combining the results with partial input data [14, 43]. As the point set can be viewed as a token sequence, PoinTr [43] employs the transformer architecture [26] to predict the missing point proxies. SeedFormer [50] introduces a shape representation called patch seeds for preventing the loss of local information during pooling operation. Some other approaches [17, 18, 46] propose to enhance the generated shapes by exploiting the structural relations in the refinement stage. However, these strategies employ a unified refinement strategy for all points, which limits their ability to generate pleasing geometric details for different points. Our approach differs from theirs by breaking down the shape refinement task into two sub-goals, and adaptively extracting reliable features for different partial areas.

2.2. Multi-view fusion for Shape Learning

View-based 3D shape recognition techniques have gained significant attention in recent years. The classic Multi-View Convolutional Neural Network (MVCNN) model was introduced in [22], where color images are fed into a CNN and subsequently combined by a pooling operation. However, this approach has the fundamental drawback of ignoring view relations. Following works [7, 30, 10, 40] propose various strategies to tackle this problem. For example, Yang *et al.* [40] obtains a discriminative 3D object representation by modeling region-to-region relations. LSTM is used to build the inter-view relations [5]. Since the cross-modal data are more available recently, methods are proposed to fuse features of views and point clouds [41, 42]. Inspired by the success of multi-view fusion, our method utilizes point cloud features to enhance relationships between multiple views obtained by self-view augmentation.

3. Method

The input of our SVDFormer consists of three parts: a partial and low-res point cloud $P_{in} \subseteq \mathbb{R}^{N\times 3}$, N_V camera locations $VP \subseteq \mathbb{R}^{N_V\times 3}$ (three orthogonal views in our experiments), and N_V depth maps $D \subseteq \mathbb{R}^{N_V\times 1\times H\times W}$. Given these inputs, our goal is to estimate a complete point cloud $P_2 \subseteq \mathbb{R}^{N_2\times 3}$ in a coarse-to-fine manner. The over-

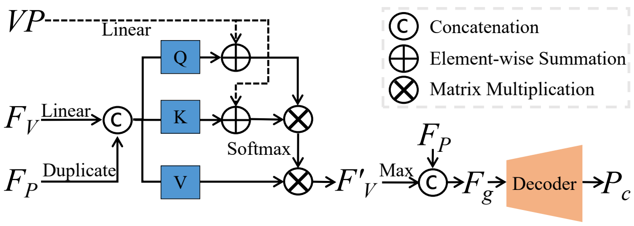

Figure 3. Illustration of the feature fusion module.

all architecture is exhibited in Figure 2, which comprises two parts: an SVFNet and a refiner equipped with two SDG modules. SVFNet first leverages multiple self-projected depth maps to produce a globally completed shape $P_0 \subseteq \mathbb{R}^{N_0 \times 3}$. Subsequently, two SDGs gradually refine and upsample P_0 to yield the final point cloud P_2, which exhibits geometric structures with high levels of detail. Note that unlike some recent cross-modal approaches [48, 51, 1, 45], *our method makes full use of self-structures and does not require any additional paired information such as color images with precisely calibrated camera intrinsic parameters [48, 51]*. Depth maps are directly yielded by projecting point clouds themself from controllable viewpoints during the data-preprocessing stage.

3.1. SVFNet

The SVFNet aims to observe the partial input from different viewpoints and learns an effective descriptor to produce a globally plausible and complete shape. We first extract a global feature F_p from P_{in} using a point-based 3D backbone network and a set of view features F_V from the N_V depth maps using a CNN-based 2D backbone network. We directly adopt well-established backbone networks. In detail, the PointNet++ [20] with three set abstraction layers encodes P_{in} in a hierarchical manner and the ResNet-18 model [11] is employed as the 2D backbone.

However, how to effectively fuse the above cross-modal features is challenging. In our early experiments, we directly concatenate these features, but the produced shape is less pleasing (see the ablation studies in Section 4.5). This may be caused by the domain gap between 2D and 3D representations. To resolve this problem, we propose a new feature fusion module, to fuse F_p and F_V, and output a global shape descriptor F_g, followed by a decoder to generate the global shape P_c. The decoder uses a 1D Conv-Transpose layer to transform F_g to a set of point-wise features and regresses 3D coordinates with a self-attention layer [26]. Finally, we adopt a similar approach to previous studies [35, 50], where we merge P_c and P_{in} and resample the merged output to generate the coarse result P_0.

Feature Fusion. As shown in Figure 3, F_V is first transformed to query, key, and value tokens via linear projection and the guidance of global shape feature F_P. Then, to enhance the discriminability of view features, the attention weights are calculated based on the query and key tokens conditioned on the projected viewpoints VP. Detailedly, we map VP into the latent space through a linear transformation and then use them as positional signals for feature fusion. After the elemental-wise product, each feature in F'_V combines the relational information from other views under the guidance of F_P. Finally, the output shape descriptor F_g is derived from F'_V via maximum pooling.

3.2. SDG

The SDG seeks to generate a set of coordinate offsets to fine-tune and upsample the coarse shape, based on the structural type of the missing surface region. To achieve it, SDG is designed as a dual-path architecture as shown in Figure 4, which consists of two parallel units named *Structure Analysis* and *Similarity Alignment*, respectively. Overall, fed with the partial input P_{in} and coarse point cloud P_{l-1} outputted in the last step, we obtain the combined point-wise feature $F_l \subseteq \mathbb{R}^{N \times 2C}$. F_l comprises two kinds of sources of shape information: one is derived from learned shape priors, while the other is learned from the similar geometric patterns found within P_{in}. F_l is then projected to a higher dimensional space and reshaped to produce a set of up-sampled offsets $O_l \subseteq \mathbb{R}^{rN \times 3}$, where r represents the upsampling rate. The predicted offsets are further added back to P_{l-1} to obtain a new completion result. Note that we iterate SDG twice, as shown in Figure 2.

3.2.1 Structure Analysis

Since detailed geometries from missing regions are harder to be recovered, we embed an incompleteness-aware self-attention layer to explicitly encourage the network to focus more on the miss regions. Specifically, P_{l-1} is first concatenated with the shape descriptor F_g, and then embedded into a set of point-wise feature $F_{l-1} = \{f_i\}_{i=1}^{N_{l-1}}$ by a linear layer. Next, F_{l-1} is fed to the incompleteness-aware self-attention layer to obtain a set of features $F_Q = \{q_i\}_{i=1}^{N_{l-1}}$, which encodes the point-wise incompleteness information. q_i is computed by:

$$\begin{aligned} q_i &= \sum_{j=1}^{N_{l-1}} a_{i,j}(f_j W_V) \\ a_{i,j} &= Softmax((f_i W_Q + h_i)(f_j W_K + h_j)^T) \end{aligned}, \quad (1)$$

where W_Q, W_K, and W_V are learnable matrix with the size of $C \times C$. h_i is a vector that represents the degree of incompleteness for each point x in P_{l-1}. Intuitively, points in missing regions tend to have a larger distance value to the partial input. We thus calculate the incompleteness by:

$$h_i = Sinusoidal(\frac{1}{\gamma} \min_{y \in P_{in}} ||x - y||), \quad (2)$$

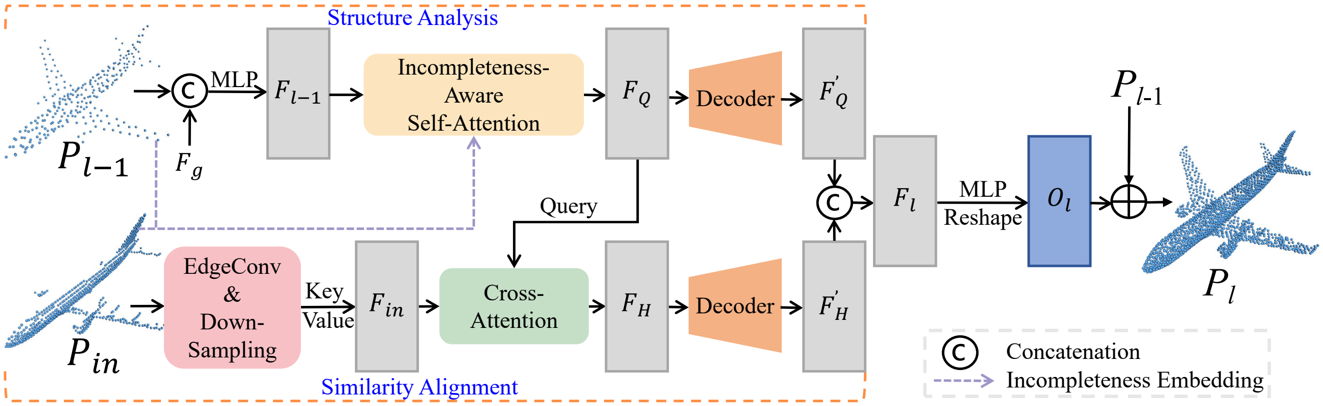

Figure 4. The architecture of SDG. The upper path represents Structure Analysis and the lower path represents Similarity Alignment. Each sub-network generates an offset feature which is then concatenated with each other and used to regress into the coordinate offsets.

where γ is a scaling coefficient. We set it as 0.2 in our experiment. The sinusoidal function [26] is used to ensure that h_i has the same dimension as the embeddings of query, key, and value. We then decode F_Q into $F_Q^{'}$ for further analysis of the coarse shape.

3.2.2 Similarity Alignment

The *Similarity Alignment* unit exploits the potential similar local pattern in P_{in} for each point in P_{l-1} and addresses the feature mismatch problem caused by the unordered nature of point clouds. Inspired by the point proxies in [43], we begin by using three EdgeConv layers [29] to extract a set of downsampled point-wise feature F_{in}. Each vector in F_{in} captures the local context information. Since there could exist long-range similar structures, we then perform feature exchange by cross-attention, which is a classical solution for feature alignment. The calculation process is similar to vanilla self-attention. The only difference lies in that the query matrix is produced by F_Q, while F_{in} serves as the key and value vectors. The cross-attention layer outputs point-wise feature $F_H \subseteq \mathbb{R}^{N_{l-1} \times C}$, which integrates similar local structures in P_{in} for each point in the coarse shape P_{l-1}. In this way, this unit can model the geometric similarity between two point clouds and facilitate the refinement of points with similar structures in the input. Similar with the structure analysis unit, F_H is also decoded into a new feature $F_H^{'}$. These two decoders have the same architecture, which is implemented with two layers of self-attention [26]. For more details of the used self-attention, cross-attention, and decoders, please refer to the supplemental files.

3.3. Loss Function

In order to measure the differences between the generated point cloud and the ground truth P_{gt}, we use the Chamfer Distance (CD) as our loss function, which is a common choice in recent works. To facilitate the coarse-to-fine generation process, we regularize the training by computing the loss function as:

$$\mathcal{L} = \mathcal{L}_{CD}(P_c, P_{gt}) + \sum_{i=1,2} \mathcal{L}_{CD}(P_i, P_{gt}) \tag{3}$$

It is worth noting that we downsample the P_{gt} to the same density as P_c, P_1, P_2 in order to compute the losses.

4. Experiment

4.1. Dataset and Evaluation Metric

We first use the PCN [44] and ShapeNet-55/34 [43] dataset for evaluation. To ensure a fair comparison, we follow the same experiment settings as previous methods [43, 50]. PCN contains shapes from 8 categories in ShapeNet [2]. The ground-truth complete point cloud has 16,384 points and the partial input has 2,048 points. ShapeNet-55 [43] is also created based on ShapeNet [2] and contains shapes from 55 categories. The ground-truth point cloud has 8,192 points and the partial input has 2,048 points. ShapeNet-34 contains 34 categories for training and leaves 21 unseen categories for testing, which is used for evaluation of generalization ability on novel categories that are unseen during training. Secondly, to evaluate the generalization ability in real-world scenarios, we test our method on both KITTI [8] and ScanNet [3], which contain partial point clouds extracted from LiDAR scans and RGB-D scans, respectively. Specifically, we test on 2,401 KITTI cars extracted by [44], and 100 chair point clouds from ScanNet. We use CD, Density-aware CD (DCD) [34], and F1-Score as evaluation metrics. Following the recent work [50], we report the ℓ^1 version of CD for PCN and the ℓ^2 version of CD for Shapenet-55, for an easier comparison.

4.2. Results on the PCN Dataset

We compare SVDFormer with state-of-the-art methods [44, 37, 28, 47, 43, 35, 33, 38, 50, 46] in Table 1. CD

Table 1. Quantitative results on the PCN dataset. (ℓ^1 CD $\times 10^3$ and F1-Score@1%)

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	CD-Avg↓	DCD↓	F1↑
PCN [44]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64	-	0.695
GRNet [37]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83	0.622	0.708
CRN [28]	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05	8.51	-	0.652
NSFA [47]	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48	8.06	-	0.734
PoinTr [43]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38	0.611	0.745
SnowflakeNet [35]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21	0.585	0.801
SDT [46]	4.60	10.05	8.16	9.15	8.12	10.65	7.64	7.66	8.24	-	0.754
PMP-Net++ [33]	4.39	9.96	8.53	8.09	6.06	9.82	7.17	6.52	7.56	0.611	0.781
FBNet [38]	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18	6.94	-	-
Seedformer [50]	3.85	9.05	8.06	7.06	**5.21**	8.85	6.05	5.85	6.74	0.583	0.818
Ours	**3.62**	**8.79**	**7.46**	**6.91**	5.33	**8.49**	**5.90**	**5.83**	**6.54**	**0.536**	**0.841**

Table 2. Quantitative results on ShapeNet-55. CD-S, CD-M, and CD-H stand for CD values under the easy, moderate, and hard difficulty levels, respectively. (ℓ^2 CD $\times 10^3$ and F1-Score@1%)

Methods	CD-S	CD-M	CD-H	CD-Avg↓	DCD-Avg↓	F1↑
FoldingNet [39]	2.67	2.66	4.05	3.12	-	0.082
PCN [44]	1.94	1.96	4.08	2.66	0.618	0.133
TopNet [24]	2.26	2.16	4.3	2.91	-	0.126
PFNet [14]	3.83	3.87	7.97	5.22	-	0.339
GRNet [37]	1.35	1.71	2.85	1.97	0.592	0.238
PoinTr [43]	0.58	0.88	1.79	1.09	0.575	0.464
SeedFormer [50]	0.50	0.77	1.49	0.92	0.558	**0.472**
Ours	**0.48**	**0.70**	**1.30**	**0.83**	**0.541**	0.451

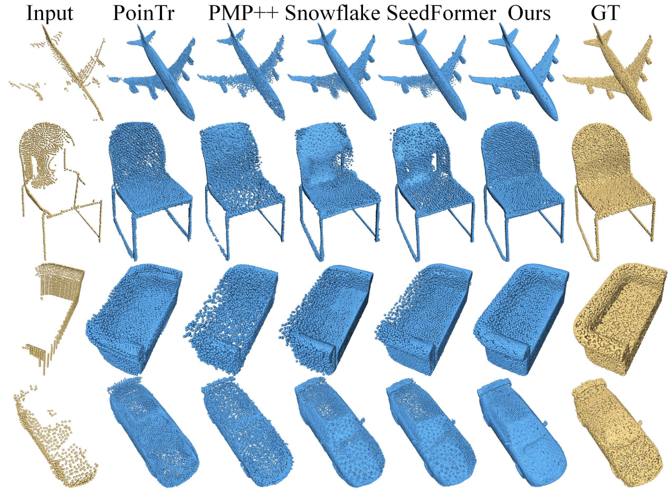

Figure 5. Visual comparisons with recent methods [43, 33, 35, 50] on the PCN dataset. Our method produces the most complete and detailed structures compared to its competitors.

values are courtesy of [50, 43], while F1-Score and DCD values are computed using their pre-trained models. The quantitative results demonstrate that SVDFormer achieves almost the best performance across all metrics. Especially, our method outperforms SeedFormer by 8.06% in DCD.

Figure 5 provides a visual comparison of the results produced by the different methods. In the case of the car and plane models, all methods are successful in generating the overall shapes. However, SVDFormer outperforms the other methods by producing sharper and more complete edges for detailed structures, such as plane wings and car spoilers. This is due to the generation ability of SDG. In the case of chair and couch models, SVDFormer can accurately locate missing regions and generate points among the holes in the models, leading to more faithful results.

4.3. Results on the ShapeNet-55/34 Dataset

The test set of ShapeNet-55 can be classified into three levels of difficulty: simple (S), moderate (M), and hard (H), which correspond to different numbers of missing points (2,048, 4,096, and 6,144). The quantitative results are presented in Table 2, consisting of CD values for three difficulty levels and the average value of two additional metrics. Our method achieves the best result in CD across all difficulty settings. Notably, SVDFormer outperforms the state-of-the-art method SeedFormer, achieving a 12.8% improvement in CD for the hard difficulty level. The results under different difficulty levels are shown in Figure 6. Compared to PoinTr and SeedFormer, our method produces smoother surfaces. The visual results clearly demonstrate that SVDFormer is capable of efficiently recovering geometries from shapes with varying degrees of incompleteness.

We further evaluate SVDFormer on the ShapeNet-34 dataset. The results on both seen and unseen categories are detailed in Table 3,which shows that SVDFormer achieves the best performance in terms of all three metrics.

Table 3. Quantitative results on ShapeNet-34.

Methods	34 seen categories						21 unseen categories					
	CD-S	CD-M	CD-H	CD-Avg↓	DCD-Avg↓	F1↑	CD-S	CD-M	CD-H	CD-Avg↓	DCD-Avg↓	F1↑
FoldingNet [39]	1.86	1.81	3.38	2.35	-	0.139	2.76	2.74	5.36	3.62	-	0.095
PCN [44]	1.87	1.81	2.97	2.22	0.624	0.150	3.17	3.08	5.29	3.85	0.644	0.101
TopNet [44]	1.77	1.61	3.54	2.31	-	0.171	2.62	2.43	5.44	3.50	-	0.121
PFNet [14]	3.16	3.19	7.71	4.68	-	0.347	5.29	5.87	13.33	8.16	-	0.322
GRNet [37]	1.26	1.39	2.57	1.74	0.600	0.251	1.85	2.25	4.87	2.99	0.625	0.216
PoinTr [43]	0.76	1.05	1.88	1.23	0.575	0.421	1.04	1.67	3.44	2.05	0.604	0.384
SeedFormer [50]	0.48	0.70	1.30	0.83	0.561	0.452	**0.61**	1.07	2.35	1.34	0.586	0.402
Ours	**0.46**	**0.65**	**1.13**	**0.75**	**0.538**	**0.457**	**0.61**	**1.05**	**2.19**	**1.28**	**0.554**	**0.427**

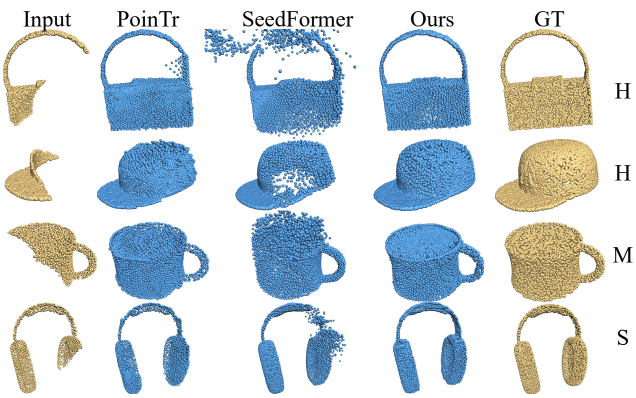

Figure 6. Visual comparison with two representative approaches [43, 50] on ShapeNet-55. H (Hard), M (Moderate), and S (Simple) stand for the three difficulty levels.

4.4. Results on Real-world Scans

For the real-world scans, as there is no ground truth available for real-world partial point clouds, we evaluate the models pre-trained on PCN without fine-tuning or re-training. We report the Minimal Matching Distance (MMD) [44] (see from Table 4) as a quantitative evaluation metric to assess the similarity of the output to a typical car/chair for the real-world scans. Also, we demonstrate the visual comparisons in Figure 7. Our method can produce cleaner shapes with detailed structures and sharp edges. It can be concluded that our method can generate more detailed results even when the input is extremely sparse and has a different distribution from the training data.

4.5. Ablation Studies and Discussions

To ablate SVDFormer, we remove and modify the main components. All ablation variants are trained and tested on the PCN dataset. The ablation variants can be categorized as ablations on SVFNet and SDG.

Ablation on SVFNet. To investigate the impact of shape descriptor extraction methods, we compare two variants of SVFNet, and the results are presented in Table 5. In the variant A, we remove the input depth maps, and the completion

Table 4. Quantitative results on Real-world Scans. All the results are produced by models pre-trained on PCN (MMD $\times 10^3$).

Methods	GRNet [37]	PoinTr [43]	SeedFormer [50]	Ours
KITTI [8]	5.350	32.854	1.179	**0.967**
ScanNet [3]	2.672	2.516	2.231	**1.926**

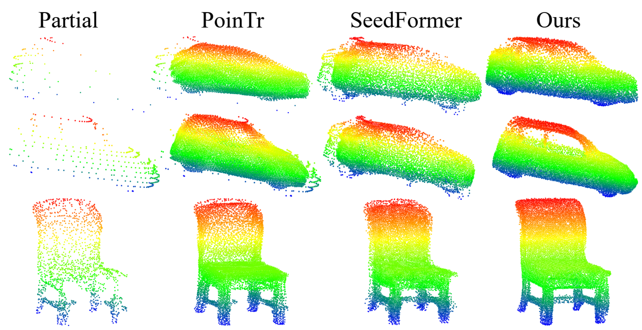

Figure 7. Visual comparison on real-world scans. All results are produced by models pre-trained on PCN.

Table 5. Effect of SVFNet. (ℓ^1 CD $\times 10^3$ and F1-Score@1%)

Methods	CD↓	DCD↓	F1↑
A : w/o Projection	6.63	0.547	0.831
B : w/o Feature Fusion	6.68	0.551	0.827
Ours	**6.54**	**0.536**	**0.841**

Table 6. Effect of SDG. (ℓ^1 CD $\times 10^3$ and F1-Score@1%)

Methods	Analysis	Alignment	Embedding	CD↓	DCD↓	F1↑
C	✓	✓		6.69	0.549	0.829
D	✓		✓	6.76	0.552	0.825
E		✓		6.78	0.556	0.823
F	✓			6.88	0.561	0.819
Ours	✓	✓	✓	**6.54**	**0.536**	**0.841**
SeedFormer [50]				6.74	0.583	0.818
SnowflakeNet-baseline [35]				7.21	0.585	0.801
SnowflakeNet + SDG				6.73	0.553	0.828

performance is limited by relying only on 3D coordinates to understand shapes. In the variant B, we evaluate the impor-

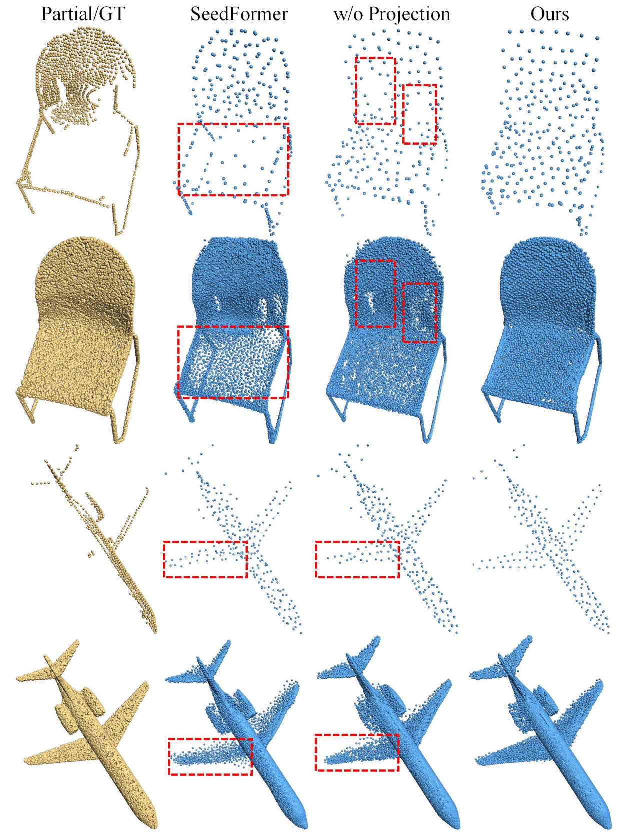

Figure 8. Visual comparison of the representative coarse-to-fine method [50] and our variant A (w/o projection) on two partial models. The upper results are the generated coarse point clouds.

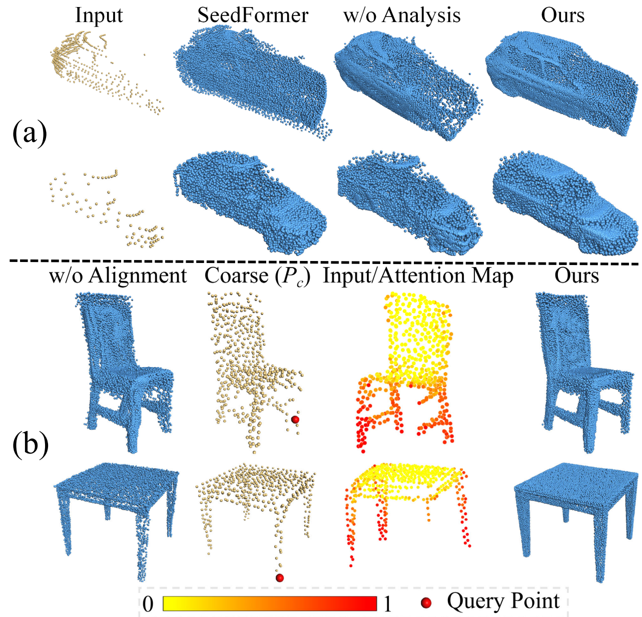

Figure 9. (a): Visual comparison of [50] and variant E (w/o Structure Analysis) on LiDAR scans. (b): Visual comparison of variant E (w/o Similarity Alignment) and generated coarse point cloud P_c on RGB-D scans. We select a query point (marked with red) in P_c and visualize the attention map in the cross-attention layer. The redder the color, the higher the similarity.

tance of our Feature Fusion module by replacing the fusion of different inputs with late fusion, which directly concatenates F_p and F_v. We observe an evident drop in performance, indicating that the proposed SVFNet can effectively fuse cross-modal features.

Furthermore, to conduct a more thorough analysis of the effectiveness of our SVFNet, we generate visualizations of the results produced by our approach, our variant A, and SeedFormer, which also employ the coarse-to-fine paradigm. In Figure 8, we present the results alongside the coarse point cloud generated directly by SVFNet (patch seeds of [50]). Our analysis reveals that during the initial coarse completion stage, both SeedFormer and the variant A produce suboptimal results, such as generating too few points in missing areas. This presents a challenge for the refinement stage, making it difficult to produce satisfactory final results. Our SVFNet overcomes this challenge by leveraging multiple viewpoints to observe partial shapes. By doing so, our method is able to locate missing areas and generate a compact global shape, leading to the production of fine details in the final results. See the supplementary for additional ablation experiments for the number of views.

Ablation on SDG. Table 6 compares different variants of SVDFormer on the SDG module. In the variant C, we remove the Incompleteness Embedding of SDG, which results in a higher CD value and a lower F1-Score, indicating that the ability to perceive the incompleteness degree of each part is crucial for the model's performance. In the variants D and E, we completely remove the Similarity Alignment and Structure Analysis path from SDG, respectively. The results show that the performance of the model drops when any one of these paths is removed.

To better understand and analyze SDG, we show more visual results in Figure 9. Specifically, we investigate the effectiveness of the Structure Analysis path and the Similarity Alignment unit by comparing the performance of different variants of the model on real-world scans. In Figure 9 (a), our method can generate plausible shapes, while the variant E and SeedFormer produce undesired structures due to over-preserving the partial input. This result proves the importance of the Structure Analysis path, particularly when the input contains limited information. In Figure 9 (b), we compare the results of our method with the variant D. We show the generated coarse shape and select a query point (missed in input) in P_c. We then visualize the attention map in the cross-attention layer to demonstrate the effectiveness of the Similarity Alignment unit. The results show that, for shapes with highly-similar regions, the Similarity Alignment unit can locate similar geometries of short or long-range distances, leading to finer details.

Table 7. Complexity analysis. We compare the inference time (ms) and the number of parameters (Params) of our method and three classical methods on ShapeNet-55. Our method achieves a balance between computation cost and performance.

Methods	Time	Params	CD↓	DCD↓
GRNet [37]	10.67ms	73.15M	1.97	0.592
PoinTr [43]	12.36ms	30.09M	1.07	0.575
SeedFormer [50]	40.63ms	3.24M	0.92	0.558
Ours	23.11ms	19.62M	0.83	0.546

Extending SDG to other methods. In addition, we evaluate the generation ability of SDG by replacing the SPD in SnowflakeNet [35] with SDG. As presented in Table 6, SnowflakeNet achieves better performance in terms of all metrics, when paired with our SDG module. This indicates that our disentangled refiner has better generation ability.

Complexity Analysis. We show the complexity analysis in Table 7, where the inference time on a single NVIDIA 3090 GPU, number of parameters, and the results on ShapeNet-55 are shown. The comparison indicates that our method achieves a trade-off between cost and performance.

5. Conclusion

We propose SVDFormer for point cloud completion. We start by identifying the main challenges in the completion and developing new solutions for each of them. SVDFormer leverages self-projected multi-view analysis to comprehend the overall shape and effectively perceive missing regions. Furthermore, we introduce a decoder called Self-structure Dual-generator that breaks down the shape refinement process into two sub-goals, resulting in a disentangled but improved generation ability. Experiments on various shape types demonstrate that SVDFormer achieves the state-of-the-art performance on point cloud completion.

Acknowledgements

The work described in this paper was supported by the National Natural Science Foundation of China (No. 62172218), the Research Grants Council of the Hong Kong Special Administrative Region, China (No. UGC/FDS16/E14/21), the Shenzhen Science and Technology Program (No. JCYJ20220818103401003, No. JCYJ20220530172403007), the Natural Science Foundation of Guangdong Province (No. 2022A1515010170), and Hong Kong RGC General Research Fund (No. 15218521).

References

[1] Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 3, 4

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5, 7

[4] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017. 2

[5] Guoxian Dai, Jin Xie, and Yi Fang. Siamese cnn-bilstm architecture for 3d shape representation learning. In *IJCAI*, pages 670–676, 2018. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 12, 13

[7] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 264–272, 2018. 3

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5, 7

[9] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. 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*, pages 85–93, 2017. 2

[10] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 12, 13

[12] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4completion: Synthesizing multi-view depth maps for 3d shape completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 3

[13] Sheng Yu Huang, Hao-Yu Hsu, and Yu-Chiang Frank Wang. SPoVT: Semantic-prototype variational transformer for dense point cloud semantic completion. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[14] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 2, 3, 6, 7

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[16] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020. 2

[17] Liang Pan. Ecg: Edge-aware point cloud completion with graph convolution. *IEEE Robotics and Automation Letters*, 5(3):4392–4398, 2020. 3

[18] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8524–8533, 2021. 2, 3

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 12

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4, 12

[21] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 2

[22] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 3

[23] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1726–1735, 2022. 2, 3

[24] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 6

[25] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017. 2

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 4, 5, 12

[27] Xiaogang Wang, Marcelo H Ang, and Gim Hee Lee. Voxel-based network for shape completion by leveraging edge generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13189–13198, 2021. 2

[28] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 2, 5, 6

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

[30] Xin Wei, Ruixuan Yu, and Jian Sun. Learning view-based graph convolutional network for multi-view 3d shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[31] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13080–13089, 2021. 12

[32] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1939–1948, 2020. 2

[33] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 5, 6, 13

[34] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Balanced chamfer distance as a comprehensive metric for point cloud completion. *Advances in Neural Information Processing Systems*, 34:29088–29100, 2021. 5

[35] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021. 2, 3, 4, 5, 6, 7, 9, 12, 13, 14

[36] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628, 2021. 3, 12, 13

[37] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European*

Conference on Computer Vision, pages 365–381. Springer, 2020. 2, 5, 6, 7, 9, 13

[38] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, pages 676–693. Springer, 2022. 2, 3, 5, 6

[39] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018. 2, 6, 7

[40] Ze Yang and Liwei Wang. Learning relationships for multi-view 3d object recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7505–7514, 2019. 3

[41] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1310–1318, 2018. 3

[42] Haoxuan You, Yifan Feng, Xibin Zhao, Changqing Zou, Rongrong Ji, and Yue Gao. Pvrnet: Point-view relation neural network for 3d shape recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9119–9126, 2019. 3

[43] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12498–12507, 2021. 1, 2, 3, 5, 6, 7, 9, 12, 13, 14, 16

[44] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 2, 5, 6, 7, 12

[45] Bowen Zhang, Xi Zhao, He Wang, and Ruizhen Hu. Shape completion with points in the shadow. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 4

[46] Wenxiao Zhang, Zhen Dong, Jun Liu, Qingan Yan, Chunxia Xiao, et al. Point cloud completion via skeleton-detail transformer. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2, 3, 5, 6

[47] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision*, pages 512–528. Springer, 2020. 2, 5, 6

[48] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15890–15899, 2021. 2, 4

[49] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10256–10265, 2021. 2

[50] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European Conference on Computer Vision*, pages 416–432. Springer, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 16

[51] Zhe Zhu, Liangliang Nan, Haoran Xie, Honghua Chen, Jun Wang, Mingqiang Wei, and Jing Qin. Csdn: Cross-modal shape-transfer dual-refinement network for point cloud completion. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2, 4

In this supplementary material, we provide more detailed information to complement the main manuscript. Specifically, we first introduce the implementation details, including network architecture details and experimental settings. Then, we conduct more ablation studies to analyze our method. Next, we provide some failure cases and a discussion on the limitations of our work. Finally, we present additional quantitative and qualitative results.

A. Detailed Settings

Network implementation details. We apply perspective projection to get the depth maps with the resolution of 224 × 224 from three orthogonal views. We directly feed the projected depth maps to the network without applying any color mapping enhancement. In SVFNet, we use PointNet++ [20] to extract features from point clouds. The detailed architecture is: $SA(C = [3, 64, 128], N = 512, K = 16) \rightarrow SA(C = [128, 256], N = 128, K = 16) \rightarrow SA(C = [512, 256])$. The final feature dimension of ResNet18 [11] is set to 256. The dimension of the embed query, key, and value in View Augment is set to 256. After concatenation, we get the shape descriptor F_g with 512 channels. We use a self-attention layer of 512 hidden feature dimensions followed by an MLP to regress the coarse points P_C. The merged point cloud P_0 has 512 and 1024 points for PCN and ShapeNet-55, respectively.

During refinement, we set the upsampling rates $\{r_1, r_2\}$ of the two SDGs as {4,8} and {2,4} for PCN and ShapeNet-55, respectively. We adopt EdgeConv [29] to extract local features from P_{in}. The detailed architecture is: $EdgeConv(C = [3, 64], K = 16) \rightarrow FPS(2048, 512) \rightarrow EdgeConv(C = [64, 256], K = 8)$. We use a shared-weights architecture above in the two SDGs. After obtaining F_Q and F_H, we use a decoder composed of two self-attention layers (one in the ShapeNet-55 experiments) to further analyze the coarse shapes. The hidden feature dimensions of self-attention layers are set as [768, $128r_1$] and [512, $128r_2$] in the two SDGs, thus producing $F_l \subseteq \mathbb{R}^{N\times 256r}$. F_l is then passed to an MLP and reshaped to $rN \times 128$. Finally, the coordinates offset is predicted by an MLP with feature dimensions of [128, 64, 3].

Usage of attention. In our method, the self-attention layer is used to generate P_c in SVFNet and decode F_Q and F_H in SDG. We also use a cross-attention layer to find the geometric similarity. In our experiments, we implement the self-attention module and the cross-attention module following the same transformer architecture [26]. The point-wise features are regarded as sequence data. The calculation procedure is illustrated in Figure 10. Given the input feature $F_{in} = \{f_i\}_{i=1}^{N_{l-1}}$, the output feature matrix $Z = \{z_i\}_{i=1}^{N_{l-1}}$ is calculated as :

$$\begin{aligned} z_i &= h_i + Linear(h_i) \\ h_i &= b_i + f_i \\ b_i &= \sum_{j=1}^{N_{l-1}} a_{i,j}(f_j W_V) \\ a_{i,j} &= Softmax((f_i W_Q)(f_j W_K)^T) \end{aligned} , \tag{4}$$

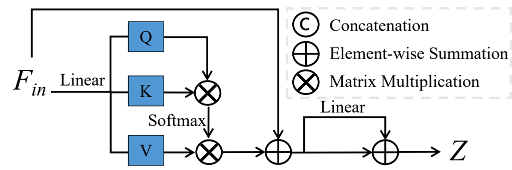

Figure 10. The calculation process of Self-Attention.

Table 8. Results and inference time of more ablation variants on PCN. (ℓ^2 CD $\times 10^3$ and F1-Score@1%)

Variants	CD↓	DCD↓	F1↑	Time
1 View	6.58	0.538	0.835	24.86ms
3 Views (Ours)	6.54	0.536	**0.841**	26.55ms
6 Views	6.55	0.536	0.840	27.11ms
Random Projection (inference)	6.58	0.537	0.838	26.25ms
Encoder in SpareNet [36]	6.66	0.551	0.825	37.26ms
ResNet-50 [11] as the 2D backbone	**6.52**	**0.535**	**0.841**	31.37ms
Vit-B/16 [6] as the 2D backbone	6.56	0.543	0.837	34.16ms

Experiment and training settings. The network is implemented using PyTorch [19] and trained with the Adam optimizer [15] on NVIDIA 3090 GPUs.

For training on the PCN dataset [44], the initial learning rate is set to 0.0001 and decayed by 0.7 for every 40 epochs. The batch size is set to 12. It takes 400 epochs for convergence. Since the point coordinates in PCN are normalized to [-0.5, 0.5], the depth maps are projected at a distance of 0.7 in order to observe the whole shape. To ensure that the input point cloud contains exactly 2048 points, we take a subset for point clouds with more than 2048 points and randomly duplicate points for those with less than 2048 points.

For training on ShapeNet-55/34 [43], the number of missing points is randomly selected from 2048 to 6192. The initial learning rate is set to 0.0001 and decayed by 0.98 for every 2 epochs. The batch size is set to 16. It takes 300 epochs for convergence. The point coordinates in ShapeNet-55 are normalized to [-1.0, 1.0]. Therefore, D are projected at a distance of 1.5. Following [35, 50], We use a partial matching strategy, which includes setting a larger resolution for P_0 and adding a partial matching loss [31].

B. Ablation Studies

Ablation on the number of projections. We conduct an ablation experiment on the number of depth maps D in

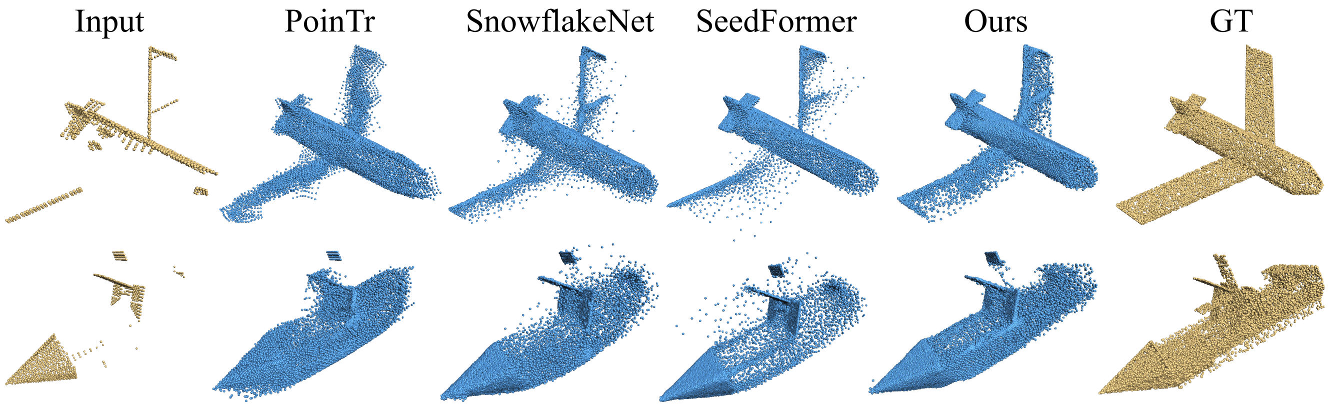

Figure 11. Example of failure cases.

Table 9. DCD results on the PCN dataset. Lower is better.

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Avg
GRNet [37]	0.688	0.582	0.610	0.607	0.644	0.622	0.578	0.642	0.622
PoinTr [43]	0.574	0.611	0.630	0.603	0.628	0.669	0.556	0.614	0.611
SnowflakeNet [35]	0.560	0.597	0.603	0.582	0.598	0.633	0.521	0.583	0.585
PMP-Net++ [33]	0.600	0.605	0.614	0.613	0.610	0.647	0.577	0.622	0.611
Seedformer [50]	0.557	0.592	0.598	0.579	0.585	0.626	0.520	0.605	0.583
Ours	**0.506**	**0.549**	**0.559**	**0.524**	**0.535**	**0.579**	**0.472**	**0.562**	**0.536**

Table 10. F1-Score@1% on the PCN dataset. Higher is better.

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Avg
GRNet [37]	0.843	0.618	0.682	0.673	0.761	0.605	0.751	0.750	0.708
PoinTr [43]	0.915	0.665	0.718	0.710	0.798	0.632	0.796	0.797	0.754
SnowflakeNet [35]	0.941	0.695	0.745	0.776	0.858	0.691	0.867	0.834	0.801
PMP-Net++ [33]	0.941	0.660	0.721	0.754	0.860	0.657	0.822	0.830	0.781
Seedformer [50]	0.950	0.700	0.753	0.803	0.885	0.712	0.884	0.850	0.818
Ours	**0.962**	**0.738**	**0.792**	**0.833**	**0.897**	**0.746**	**0.901**	**0.863**	**0.841**

SVFNet. The depth maps D are projected from 1, 3, and 6 orthogonal views, respectively. The results on PCN are shown in Table 8. To balance the trade-off between effectiveness and computational consumption, we conduct all experiments using three views. This choice allows us to capture sufficient information from the point clouds while keeping the computational cost manageable.

Ablation on choice of coordinate systems. To testify the robustness of our method, during inference, we introduced random variations to the projection, including camera view angle offsets ranging from 0 to 10 degrees and observation distance displacements ranging from 0 to 0.1. The result reported in the 5th row of Table 8 shows that the performance will not significantly drop with random projections.

Ablation of different encoders. We testify the design of encoder to further demonstrate the effect of our self-view fusion feature extractor. We first replace the SVFNet with the encoder in SpareNet [36], which contains layers of channel-attentive EdgeConv, to re-produce the shape descriptor F_g. We report the new results in Table 8, which demonstrates that our self-view fusion feature extractor achieves better performance than existing encoder while having a tolerable computation cost. In addition, we ablate the choice of 2D backbone in the SVFNet. To be specific, We replace it with ResNet-50 [11] and the vision transformer (ViT-B/16) [6], respectively. We find that a larger CNN-based 2D backbone can slightly improve the performance while introducing more computation cost. Moreover, using the ViT results in unsatisfactory performance. This could be attributed to the fact that the entire model was trained from scratch, and larger models may not perform optimally with a limited amount of 3D training data.

C. Failure Cases and Limitations

Figure 11 displays the failure cases we observed. It's worth noting that in cases where input shapes lack irregular structures that are uncommon in the training dataset (such as the water wheel of a watercraft), the network may not be capable of producing satisfactory results. Nevertheless, our method still outperforms state-of-the-art (SOTA) methods [43, 35, 50] when dealing with simple geometric structures, like the body of the watercraft. Our SDG incorporates a Structure Analysis unit that leverages learned priors to complete shapes. However, its effectiveness may be constrained by the limited amount of available training data. Given that transformers have demonstrated effectiveness in scenarios with abundant training data, pretraining with large-scale 2/3D datasets could be a promising approach to address this limitation.

D. Additional Results

More detailed quantitative results for individual cases are available in Tables 9 and 10. Our method achieves the best DCD and F1-score on each category of the PCN dataset. In addition, we show more visual results in Figures 12, 13, and 14. In Figure 12, we present two partial point clouds on each category of the PCN dataset. In Figure 13, we present six partial point clouds of ShapeNet-55 and compare the results with two representative approaches [43, 50]. Our method generates more compact overall shapes and richer details. Also, we visualize more results in Figure 14, where the partial shapes are generated from two different viewpoints.

Figure 12. Visual results on the PCN dataset.

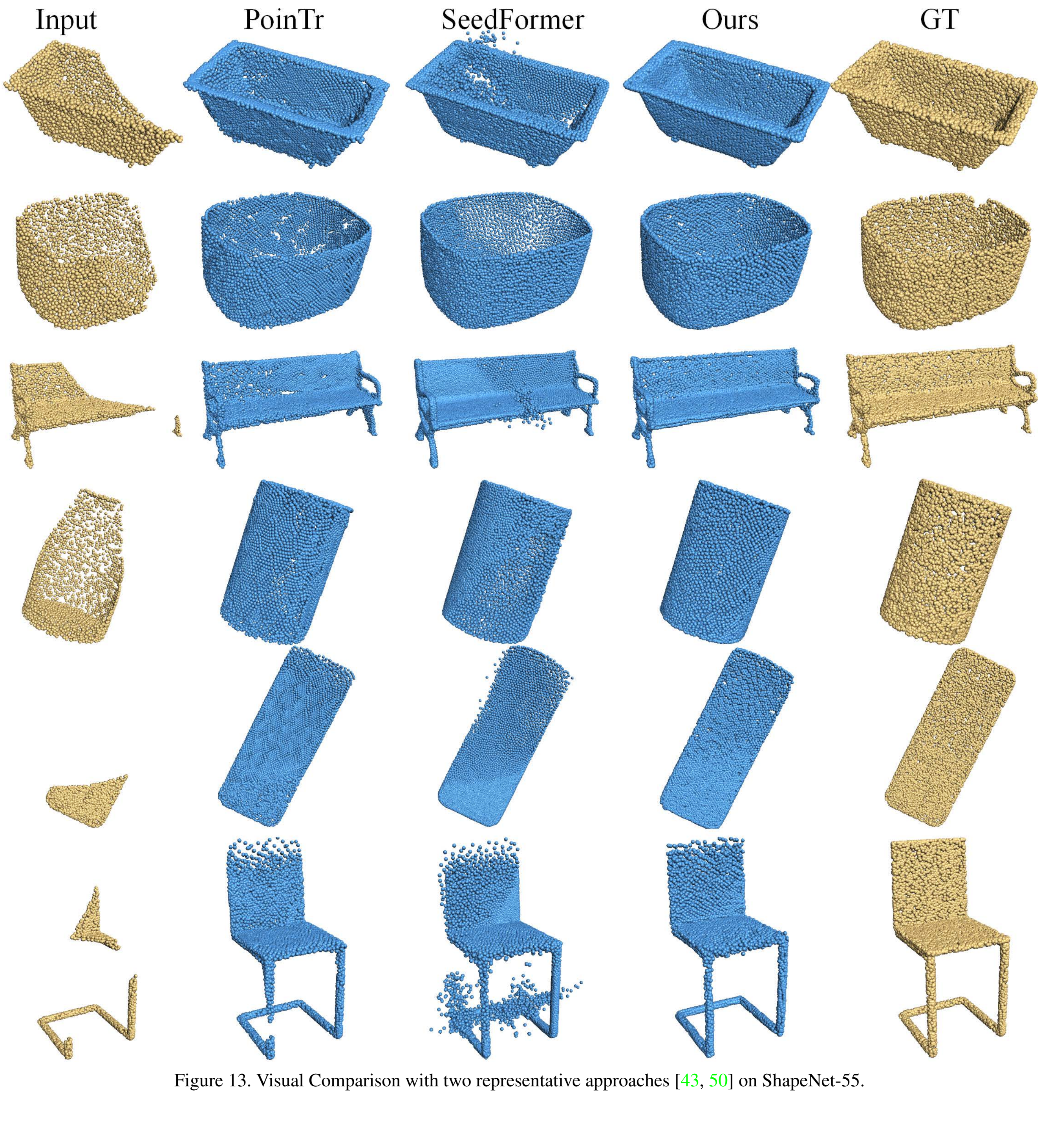

Figure 13. Visual Comparison with two representative approaches [43, 50] on ShapeNet-55.

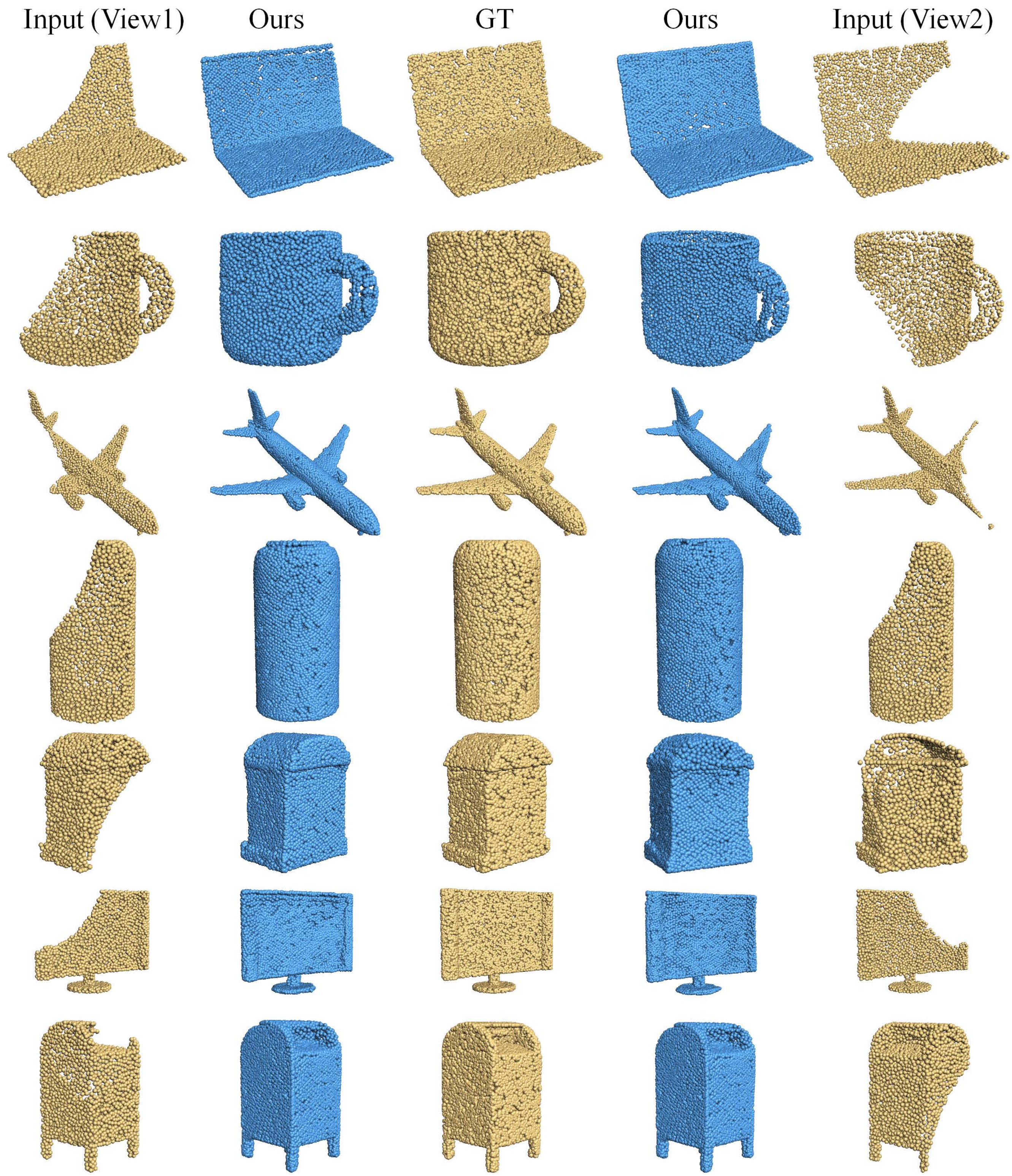

Figure 14. More visual results on ShapeNet-55. We show results when the partial input are generated from two viewpoints.