View-Guided Point Cloud Completion

Xuancheng Zhang[1], Yutong Feng[1], Siqi Li[1] Changqing Zou[3], Hai Wan[1],
Xibin Zhao[1] *, Yandong Guo[4], Yue Gao[1, 2] *

[1]BNRist, KLISS, School of Software, Tsinghua University, China
[2]THUICBS, Tsinghua University, [3]Huawei Technologies Canada Co., Ltd, [4]OPPO Research Institute

{zxc19,fyt19,lsq19}@mails.tsinghua.edu.cn, aaronzou1125@gmail.com
{wanhai,zxb,gaoyue}@tsinghua.edu.cn, yandong.guo@live.com

Abstract

This paper presents a view-guided solution for the task of point cloud completion. Unlike most existing methods directly inferring the missing points using shape priors, we address this task by introducing ViPC (view-guided point cloud completion) that takes the missing crucial global structure information from an extra single-view image. By leveraging a framework that sequentially performs effective cross-modality and cross-level fusions, our method achieves significantly superior results over typical existing solutions on a new large-scale dataset we collect for the view-guided point cloud completion task.

1. Introduction

Point cloud has attracted increasing research interest due to its wide range of applications in various fields such as auto-driving [16], robotics [27], geography [29], phenomics [19], and archaeology [4]. In practice, the point cloud's data quality directly acquired by the depth scanning devices can be affected by many factors such as occlusions between objects and low scanning precision, which may lead to the poor-quality point cloud, *e.g.*, incomplete, sparse, and noisy point cloud.

Existing methods, mainly including point cloud completion [42], denoising [5], and super-resolution (up-sampling) [38], have been proposed for the task of point cloud enhancement. Early methods generated enhanced point cloud by mainly using shape prior information [17] or hand-crafted geometric regularities [35]. In recent years, data-driven methods, especially deep learning techniques like PointNet [28] and DGCNN [37], have made significant progress on this problem. Compared to traditional methods, these deep learning based methods have demonstrated significant advantages in processing objects with irregular structure and geometry.
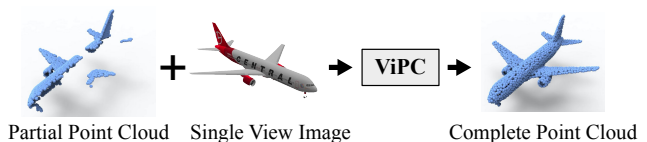

Figure 1. ViPC is a new approach completing a partial point cloud by leveraging the complementary information from an extra single-view image.

In this paper, we focus on the following point completion task: the input point cloud is incomplete but with limited noise, while our method outputs a complete point cloud. Studying this problem addresses a common problem in real-world 3D data acquisition where a 3D scanner with a RGB camera is occluded by other objects in the environment. The most recent solutions to this problem are data-driven, leveraging an encoder-decoder architecture [9, 42, 24, 39]. In those methods, an encoder transfers the incomplete input point cloud into the feature space, and then a decoder reconstructs a complete point cloud by transferring the features back to Euclidean space. The whole network works as a parameterized model by learning a mapping between the two latent spaces of incomplete and complete point cloud. In the cases where there is a large degree of incompleteness in the input point cloud, learning this mapping with only the single-modality point cloud data is challenging because of the following factors: 1) there is a great uncertainty in inferring the missing points due to the limited amount of information available, 2) point cloud is of an unstructured data, together with inherent sparseness, it is difficult to determine whether a blank 3D space is caused by inherent spareness or incompleteness.

In this paper, we seek a more applicable solution to the point cloud completion task. Specifically, we address the task with the help of the image modality and propose a view-guided point completion framework (ViPC) as illustrated in Figure. 1. This setting of sensor fusion is increasingly common as the hardware cost decreases (e.g., the In-

---
*Corresponding authors

tel Real Sense D455 and Microsoft Kinect devices). The key challenge of solving this problem is how to effectively fuse the information of pose and local details provided by the partial point cloud and the global structure information provided by the single-view image. This is not trivial because it involves a two-dimensional challenge: "cross-modality" (the information is from both image and point cloud modalities) and "cross-level" (local details and global structure are information from different levels). We address the problem by a three-stage framework which first address the cross-modality challenge and then the cross-level challenge. Specifically, the cross-modality challenge is addressed by reconstructing a coarse point cloud from the single-view image and transferring all the information required by the completion to the identical point cloud domain. The cross-level challenge is addressed by a differential refinement strategy empowered by a network called "Dynamic Offset Predictor" which can deferentially refine the points in a coarse point cloud: performing a light refinement for low-quality points while a heavy refinement for high-quality points.

To better investigate the problem, we have built a large-scale dataset called ShapeNet-ViPC on existing ShapeNet dataset [6]. Our dataset includes 38,328 objects from 13 categories. Each object has 24 sets of ground-truth data consisting of two incomplete point clouds produced under two typical data acquisition scenarios, a view-aligned image and a complete ground-truth point cloud. Extensive evaluations on ShapeNet-ViPC demonstrate that the proposed approach can achieve significantly superior results than existing state-of-the-art approaches.

In summary, the main contributions of our methods are threefold:

1. We propose a new solution for point cloud completion, in which an extra single-view image explicitly provides the crucial global structural prior information for completion.

2. We design a new general deep network for point cloud refinement which can deferentially refine the points in a point cloud.

3. We build a large-scale dataset for the point cloud completion task on the ShapeNet dataset. This dataset simulates point cloud defects caused by various kinds of occlusions. It could be used as a benchmark for future research of point cloud completion.

2. Related Works

Existing point cloud completion methods can be generally classified into three types of approaches: geometry-based, alignment-based and learning-based.

Geometry-based Methods. Geometry-based methods predict the invisible shape part from the observed shape part directly by prior geometric assumptions [20, 14]. More specifically, some methods fill the surface holes locally by generating smooth interpolations [1, 34], such as Laplacian smoothing [23] and Poisson surface reconstruction [17]. Other methods detect the regularities in model structures and repeat them to predict missing data based on the identified symmetry axes [21, 26, 35], These methods infer the missing data directly from the observed region and show impressive results. However, they need the hand-crafted geometric regularities that are predefined for specific kinds of models and only applied to models with a small degree of incompleteness.

Alignment-based Methods. Alignment-based methods retrieve identical models similar to the target object in a shape database, then align the input with temple models and complete the missing region. Some methods retrieve 3D shapes directly, such as the whole model [25] or part of them [15, 18]. Other methods use synthesized models after deformation [30] or non-3D geometric primitives such as planes and quadrics [7, 31, 41] in place of 3D shapes in the database. These methods are applicable to many different types of models and can be applied to varying degrees of incompleteness, but they require expensive cost during inference optimization and database construction; also they are sensitive to noise.

Learning-based Methods. Learning-based methods construct a parameterized model to learn a mapping between the two feature spaces of the incomplete and complete point cloud of a shape. Most of them are encoder-decoder based neural networks. As for shape representation, most existing models use voxels to represent a shape [9, 13], which are intuitive and convenient for 3D convolution. In order to preserve more geometric information (i.e., local geometric details) in the completed point cloud, several models perform the operation on point sets directly [42, 40, 33]. Since both points or voxels are a mono-modality input, it is difficult to infer an accurate mapping between an incomplete point cloud with a large-scale incompleteness and a complete point cloud. Therefore, those methods may perform well only on specific categories of objects or the shapes with a small-scale incompleteness. The work leveraging auxiliary data to supplement the missing information of the input point cloud for the enhancement task has rarely been studied.

3. Methods

3.1. Overview

Problem Definition. The proposed view-guided completion solution is based on an assumption that the input image contains the necessary structural information of the
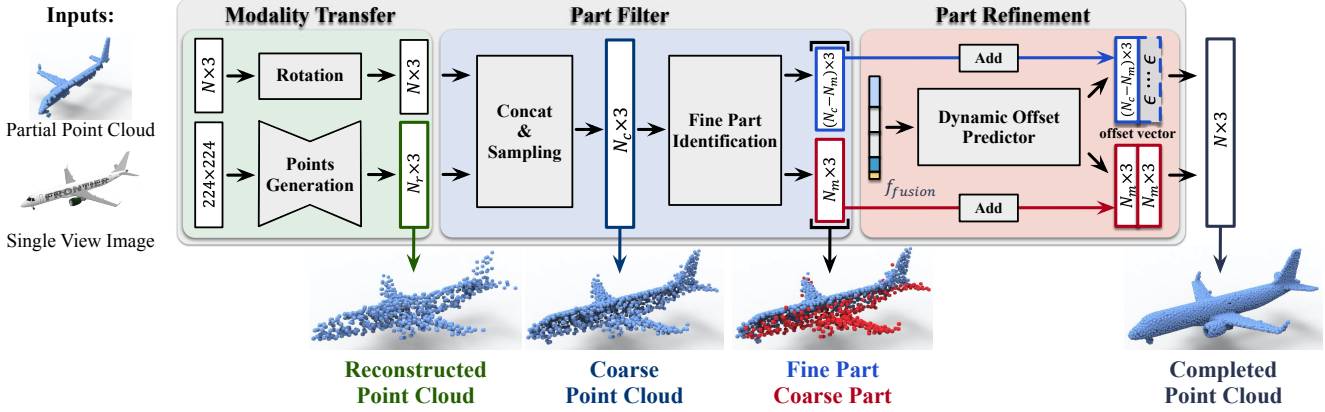
Figure 2. Architecture of the proposed ViPC; see the Pipeline section in the text for details.

missing shape part. Our goal is to recover a 3D shape $\mathcal{S}$ consisting of two parts, i.e., $\mathcal{S} = \{\hat{\mathcal{S}}_0, \Delta\mathcal{S}\}$, where $\hat{\mathcal{S}}_0$ is a shape part that is close to the input partial shape $\mathcal{S}_0$ (i.e., the difference between $\hat{\mathcal{S}}_0$ and $\mathcal{S}_0$ is limited), and $\Delta\mathcal{S}$ is the unknown missing shape part. Formally, we denote $\mathcal{M}(\mathcal{S})$ as the representation of shape $\mathcal{S}$ in modality $\mathcal{M}$, i.e. $\mathcal{M} = \mathcal{P}$ for point cloud and $\mathcal{M} = \mathcal{I}$ for image. The input of the task could be formulated as:

- The partial point cloud $\mathcal{P}(\mathcal{S}_0)$.

- The single-view image $\mathcal{I}(\mathcal{S}_I)$, where $\mathcal{S}_I$ refers to the observed shape from the view of image.

Pipeline. The proposed three-stage framework is shown in Figure 2. The first stage is used to address the cross-modality fusion problem. It maps $\mathcal{I}$ to a coarse representation of point cloud in $\mathcal{P}$. The second stage generates a coarse point cloud and the last stage enhances it and produce a higher-quality completed point cloud. These two stages work together to perform the cross-level fusion.

Specifically, the first stage termed as *Modality Transfer* maps $\mathcal{I}$ to a coarse representation of point cloud in $\mathcal{P}$ and then aligns the reconstructed point cloud to the input partial shape $\mathcal{S}_0$ in 3D space. Then the second stage termed as *Part Filter* generates a coarse point cloud from the two aligned point cloud. Additionally, this stage also makes a distinction between the points which are mainly from the input partial point cloud and the other points which are mainly reconstructed from input image $\mathcal{I}(\mathcal{S}_I)$. We term the former as the *Fine Part* and the latter as the *Coarse Part*. In general, the Fine part has much higher shape quality than the Coarse part and needs only a light refinement. Lastly, the third stage termed as *Part Refinement* takes as input both the Coarse and Fine parts and produces a completed higher-quality point cloud. In this stage, we mainly refine the shape of the Coarse but use the Fine part as the constraint for the refinement. This can achieve better results than a global refinement which refines all the points without any constraint
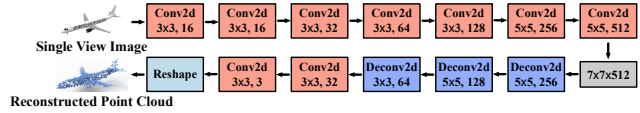

Figure 3. Network structure for point cloud generation from a single view image in Modality Transfer.

as confirmed by the ablation study in the experimental section. We achieve this by a novel neural network called "Dynamic Offset Predictor". Next, we detail the above three stages.

3.2. Modality Transfer

It would be challenging to directly reconstruct a high-quality point cloud which preserves rich local details from the image $\mathcal{I}(\mathcal{S}_I)$. Thus, the transferred point cloud is expected to work as an initialized shape mainly used for structure guidance. Let the point cloud reconstructed from the image $\mathcal{I}(\mathcal{S}_I)$ as $\mathcal{P}_r(\mathcal{S}_I)$. A light-weight point cloud reconstruction network with an encoder-decoder architecture shown in Figure 3 is employed. The encoder maps the input image into a latent space vector, and the decoder outputs an $N_r \times 3$ matrix, each row of which represents the Cartesian coordinates of a point. In our implementation, the encoder comprises a series of convolutional layers with ReLU activation, and outputs feature map in $7 \times 7 \times 512$ as the latent space vector. The decoder applies a series of deconvolutional layers and flatten the output to generate N_r point coordinates. Feature maps from each layer of the encoder are also preserved as additional guidance for the following processes in this stage. After reconstructing $\mathcal{P}_r(\mathcal{S}_I)$ from the single-view image, we align it with the input partial point cloud.

3.3. Part Filter

The reconstructed $\mathcal{P}_r(\mathcal{S}_I)$ can roughly describe the view-observed shape $\mathcal{S}_I$, which is assumed to contain the major information of the missing shape part $\Delta\mathcal{S}$. In this stage, we firstly merge $\mathcal{P}(\mathcal{S}_0)$ and $\mathcal{P}_r(\mathcal{S}_I)$ and extract a

subset (N_c points) of it as a coarse representation of shape $\mathcal{S}_0 \cup \mathcal{S}_I$. Since point density may differ in $\mathcal{P}(\mathcal{S}_0)$ and the reconstructed $\mathcal{P}_r(\mathcal{S}_I)$ and the concatenation leads to redundant points in the overlap space between $\mathcal{S}_0$ and $\mathcal{S}_I$. We would like this coarse point cloud with N_c points to be uniformly dense and preserve the global structure but also as many local details of the shape as possible. Specifically, we use the farthest point sampling (FPS) [22]) on $\mathcal{P}_c(\mathcal{S}_0 \cup \mathcal{S}_I)$ to achieve this goal.

After that, we identify the points that do not need heavy refinement as the Fine part and leave the remaining points as the Coarse part, i.e., dividing the coarse point cloud with N_c points into the Coarse part (N_m points) and Fine part ($N_c - N_m$ points). To search $N_c - N_m$ points that are close to the points in $\mathcal{P}(\mathcal{S}_0)$ for the Fine part, we construct the correspondence between points in $\mathcal{P}(\mathcal{S}_0)$ and $\mathcal{P}_c(\mathcal{S}_0 \cup \mathcal{S}_I)$ by Chamfer Distance (CD) [11]. For each point $\mathbf{p}$ in $\mathcal{P}_c(\mathcal{S}_0 \cup \mathcal{S}_I)$, we calculate its distance to the closest point in $\mathcal{P}(\mathcal{S}_0)$, i.e. $d(\mathbf{p}) = \min_{\mathbf{q} \in \mathcal{P}(\mathcal{S}_0)} \|\mathbf{p} - \mathbf{q}\|_2^2$. If the distance $d(\mathbf{p})$ is less than an adaptive threshold d_{thr}, we select it as a candidate point in the the Fine part. In the implementation, the threshold d_{thr} can adapt to the density varying of point cloud. Technically, we randomly divide the points from $\mathcal{P}_c(\mathcal{S}_0 \cup \mathcal{S}_I)$ into two subsets, and calculate the closest distance $d(\mathbf{p})$ of all points between the two subsets. Then the average distance value could be an estimation of the point cloud density and it serves as the threshold d_{thr}.

3.4. Part Refinement

Part Refinement stage further refines (up-samples) the coarse point cloud consisting of the Fine and Coarse parts to produce a complete point cloud. To achieve an effective refinement, besides of the coarse point cloud itself (represented by the point coordinates $[x, y, z]$), the refinement also leverages the guidance of four other types of features which can be obtained from the stages of Modality Transfer and Part Filter as shown in Figure 4. Those four types of features which can be classified into 2D- or 3D- guidance, together with the point coordinates $[x, y, z]$ of the coarse point cloud, are concatenated into a global feature vector shared by each points in the coarse point cloud. The proposed Dynamic Offset Predictor network takes N_c repeated global feature vectors as input and predicts the coordinate offset value for $N_c \times R$ (R defines the up-sampling rate) output points.

3D Guidance. In recent learning-based completion methods, it has been verified that completion based on the existing shape and prior knowledge is a feasible way. For example, the symmetry of airplane could help us to complete the missing wing in one side. Thus, we extract global features by applying the common encoder PointNet [28] on the partial point cloud $\mathcal{P}(\mathcal{S}_0)$ and the reconstructed $\mathcal{P}_r(\mathcal{S}_I)$.

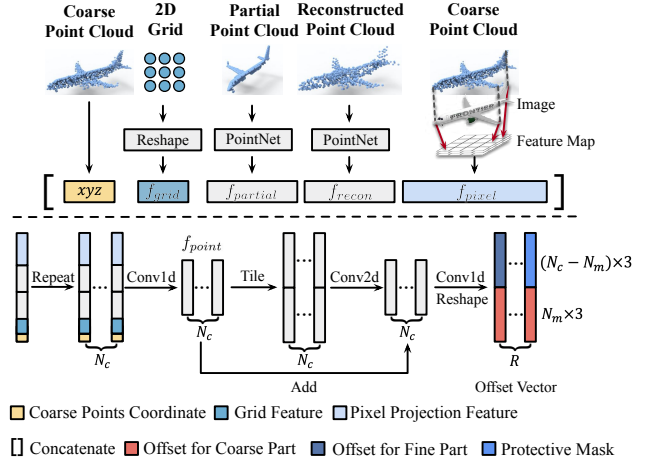**2D Guidance.** The single-view image $\mathcal{I}(\mathcal{S}_I)$ serves the

Figure 4. We concatenate five types of features (top row) as the input of Dynamic Offset Predictor whose architecture is shown in the bottom row.

guidance for the recovery of both the structure and geometrical details of the Coarse part. During the stage of modality transfer, the feature of the image $\mathcal{I}(\mathcal{S}_I)$ has been extracted into feature maps with different sizes, e.g. 56×56, 28×28, 14×14, 7×7. Inspired by the perceptual feature pooling in Pixel2Mesh [36], for each point, we search the features in the image feature maps corresponding to each point in the coarse point cloud using the coordinates and camera parameters. These features are stacked as f_{pixel} for a kind of guidance feature. In addition, to increase variations among the local points and prevent predicting the same offset for different points, inspired by FoldingNet [40], we generate a 2D-grid and reshape to a feature vector as f_{grid} to boost slight disturbance.

Dynamic Offset Predictor. Unlike the previous folding-based or tree structure methods, Dynamic Offset Predictor predicts the spatial offset of each point towards its current position rather than directly predicting the coordinates, which simplifies the regression. The data flow of Dynamic Offset Predictor shown in Figure 4 is as follows: **1.** The f_{fusion} is repeated N_c times and fed into a series of 1D convolutional layers to output the hidden embeddings f_{point}. **2.** The f_{point} is tiled R times for point movements and up-sampling. **3.** The 2D convolutional layers with kernel size $1 \times R$ are applied to perceive the local information of each point with R offsets. **4.** The offset vector in $R \times N_c \times 3$ is predicted by the 1D convolutional layers. **5.** A protective mask in $1 \times (N_c - N_m) \times 3$ with a small offset value ϵ is generated to cover the offset vector, aiming to limit the movement of Fine Part's points. The coordinates of $\mathcal{P}_{coarse}$ are then stacked R times ($R \times N_c$) and added with corresponding offsets as the completed point cloud. Therefore, *Part Refinement* with *Dynamic Offset Predictor* refines and up-samples $\mathcal{P}_{coarse}$ without changing the Fine Part's detail.

3.5. Loss Function

The loss function measures the difference between dense point cloud and ground truth. Since the point cloud are unordered data, the loss function must be permutation invariant, Chamfer Distance (CD) and Earth Mover's Distance (EMD) is considered in this work. Given two subsets $P \subseteq \mathbb{R}^3$ and $Q \subseteq \mathbb{R}^3$, Chamfer distance calculates the average closest point distance between P and Q. We use the symmetric version of CD as formulated in (1), where the first term forces the output point cloud to move close to ground truth, and the second term ensures that the output point cloud covers the ground truth point cloud.

$$\mathcal{L}_{CD} = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p-q\|_2^2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|p-q\|_2^2 \quad (1)$$

The Earth Mover's Distance (EMD) is an algorithm to evaluate the dissimilarity between two multi-dimensional distributions. Let $\phi : P \to Q$ be a bi-jection between two point clouds, which finds the minimal average distance between two point sets. In practice, searching the optimal ϕ is computationally expensive, we thereby use an iterative $(1 + \epsilon)$ approximation scheme as [2].

$$\mathcal{L}_{EMD} = \min_{\phi:P \to Q} \frac{1}{|P|} \sum_{p \in P} \|p - \phi(p)\|_2^2 \quad (2)$$

We combine these two distance with $\mathcal{L} = \alpha \mathcal{L}_{CD} + \beta \mathcal{L}_{EMD}$ as the loss function, where α, β are the trade-off hyper-parameters. We use $\alpha = 1$ and $\beta = 0.0001$ as a default value in the training stage.

4. Experimental Settings

4.1. ShapeNet-ViPC

To simulate the defects related to the task of ViPC and evaluate the performance of the proposed approach, we build a new dataset called ShapeNet-ViPC based on the ShapeNetRendering[6]. It contains 38,328 objects from 13 categories, *i.e.*, airplane, bench, cabinet, car, chair, monitor, lamp, speaker, firearm, sofa, table, cellphone, watercraft. For each object, we generate two types incomplete point cloud (with or without noise) under 24 view points, as illustrated in Figure 5. The 24 view points follow the same view point setting as ShapeNetRendering [6] (this setting is also used in 3D-R2N2 [8], PointSetGeneration [11] and Pixel2Mesh [36]). For each set we uniformly sample 2,048 points from the mesh surface of a target shape under the corresponding view point setting as the ground-truth complete point cloud. Specifically, each 3D shape is normalized into the bounding sphere with radius of 1, and rotated to the pose corresponding to a specific view point lastly. For the image data, we use the same 24 rendered views as ShapeNetRendering. ShapeNet-ViPC contains $38,328 \times 24 = 919,872$

Self occlusion + Inter-object occlusion

Scanner

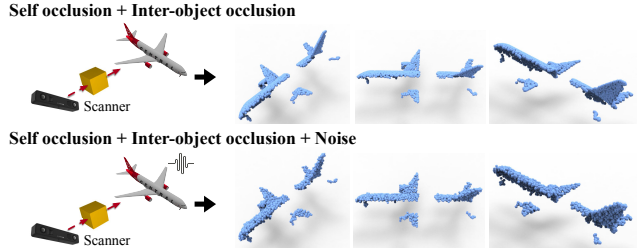Self occlusion + Inter-object occlusion + Noise

Scanner

Figure 5. Illustration of two typical types of point cloud acquisition scenarios. Top: the target shape is occluded by other objects in the environment as well as a part of itself (self-occlusion); bottom: besides of self- and inter-object occlusion, the locations of the acquired points are disturbed because of device noise.

sets of training data in total. Each set contains one ground-truth complete point cloud, two incomplete (partial) point clouds, and an image view. In this paper, we use 31,650 objects (759,600 sets) of eight categories for all experiments, 80% for training and 20% for test.

4.2. Implementation details and evaluation metrics

In our implementation, the size of input image is 224×224, from which we reconstruct a point cloud with $N_m = 784$ points. The partial input contains 2,048 points. We sample a coarse point cloud with $N_c = 1024$ points by FPS [22] from the combination of the input partial point cloud and the reconstructed point cloud (i.e., $\mathcal{P}_c(\mathcal{S}_0 \cup \mathcal{S}_I)$). The output complete cloud contains 2,048 points (i.e., $R = 2$). The network for modality transfer is pre-trained with batch size of 64 and learning rate of 1e-4 for 100 epochs. The Part Refinement network is trained with batch size of 1 and learning rate 1e-6 for 200 epochs. Category-specific parameters are trained for each object category. To quantify the completion performance, we use both the Chamfer Distance (CD) [32] and F-Score [32] as the quantitative evaluation metrics. Results with lower CD value and/or higher F-Score correspond to better completion quality.

5. Experimental Results and Analysis

5.1. Comparisons

We compare our method with several state-of-the-art methods on the task of point cloud completion, including AtlasNet [12], FoldingNet [40], point completion network (PCN) [42] and TopNet [33]. AtlasNet recovers a complete point cloud by estimating a collection of parametric surface elements. FoldingNet, 2D-grid based auto-encoder, is a pioneer grid based method for point cloud completion. PCN is an encoder-decoder framework which completes a partial input point cloud with a typical coarse-to-fine scheme. TopNet completes an imperfect point cloud by a tree structure network. All the above baseline methods take as input only a partial point cloud.

Methods	Mean Chamfer Distance per point								
	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
AtlasNet [12]	6.062	5.032	6.414	4.868	8.161	7.182	6.023	6.561	4.261
FoldingNet [40]	6.271	5.242	6.958	5.307	8.823	6.504	6.368	7.080	3.882
PCN [42]	5.619	4.246	6.409	4.840	7.441	6.331	5.668	6.508	3.510
TopNet [33]	4.976	3.710	5.629	4.530	6.391	5.547	5.281	5.381	3.350
Ours	**3.308**	**1.760**	**4.558**	**3.138**	**2.476**	**2.867**	**4.481**	**4.990**	**2.197**

Table 1. Quantitative results on ShapNet-ViPC using Chamfer Distance with 2,048 points. The best results are highlighted in bold.

Methods	F-Score@0.001								
	Avg	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft
AtlasNet [12]	0.410	0.509	0.304	0.379	0.326	0.426	0.318	0.469	0.551
FoldingNet [40]	0.331	0.432	0.237	0.300	0.204	0.360	0.249	0.351	0.518
PCN [42]	0.407	0.578	0.270	0.331	0.323	0.456	0.293	0.431	0.577
TopNet [33]	0.467	0.593	0.358	0.405	0.388	0.491	0.361	0.528	0.615
Ours	**0.591**	**0.803**	**0.451**	**0.5118**	**0.529**	**0.706**	**0.434**	**0.594**	**0.730**

Table 2. Quantitative results on ShapNet-ViPC using F-Score with 2,048 points. The best results are highlighted in bold.

Quantitative results. We normalize the output point clouds produced by the comparison methods and calculate the CD and F-Score on the 2,048 points of each shape. The results on each categories and the average are summarized in Tables 1 and 2. It is found that the proposed method consistently outperforms other methods with a significant margin on all the eight categories on both CD and F-Score metrics. Besides, our method demonstrates more advantages on the categories of airplane, watercraft, lamp, and car.

Qualitative results. We also visualize the results produced by the comparison methods for a more comprehensive evaluation. Results on the representative examples from the eight categories are shown in Figure 6. It is easy to observe that the completed point clouds produced by FoldingNet are relatively messy. The generated point clouds do not show clear structures on some shape parts, e.g., the wings of the airplane, the chair legs, the table legs. PCN and AtlasNet produce improved qualitative results compared to FoldingNet overall. However, local small-scale structural details are still missing (e.g., the fuel tank of the airplanes, the arms of the chairs) in the results. The structured-tree based TopNet method achieves better visual results than PCN and AtlasNet in general. We can see the evidences on the airplane, lamp, sofa, and watercraft, which exhibits much clearer part structures and points arranged more neatly. However, some part details in the input partial point clouds are not preserved in the completed point clouds, e.g., the points of the fuel tank of the airplane and the trestle of the table have been moved to other parts. Results produced by our method have no this problem and

show visually better performance on all eight categories than baselines. This is because unlike other comparison methods inferring the locations of all points, our method instead uses the points in the other part of the shape (i.e., Fine part) as a completion constraint and infers a local point distribution of a part of the shape (i.e., Coarse part). This makes the network preserve the local detail from the input partial point cloud, produce a more reasonable completion for the missing structure (see the reconstructed left trestles of the table), as well as converge much faster.

5.2. Experimental Analysis

Ablation Studies. We conduct ablation experiments to study the individual contributions of each stage in our method. As shown in Table 3, we quantitatively compare the point cloud qualities of the reconstructed point clouds P_{rec} generated by Modality Transfer, the coarse point clouds P_{coarse} generated by Part Filter, and the completed point clouds $P_{complete}$ generated by Part Refinement. In addition, we also study how much the differential refinement strategy that differentiates between the Fine and Coarse parts contributes. Specifically, we modify the Dynamic Offset Predictor network and disable the constraint that the points in the Fine part should be moved within a small sphere, i.e., the points from both the Fine and Coarse parts are processed without any difference. We term the results produced by this architecture as P_{global}. By comparing with P_{global} and $P_{complete}$, we can obtain the quantitative contribution of the differential refinement strategy.
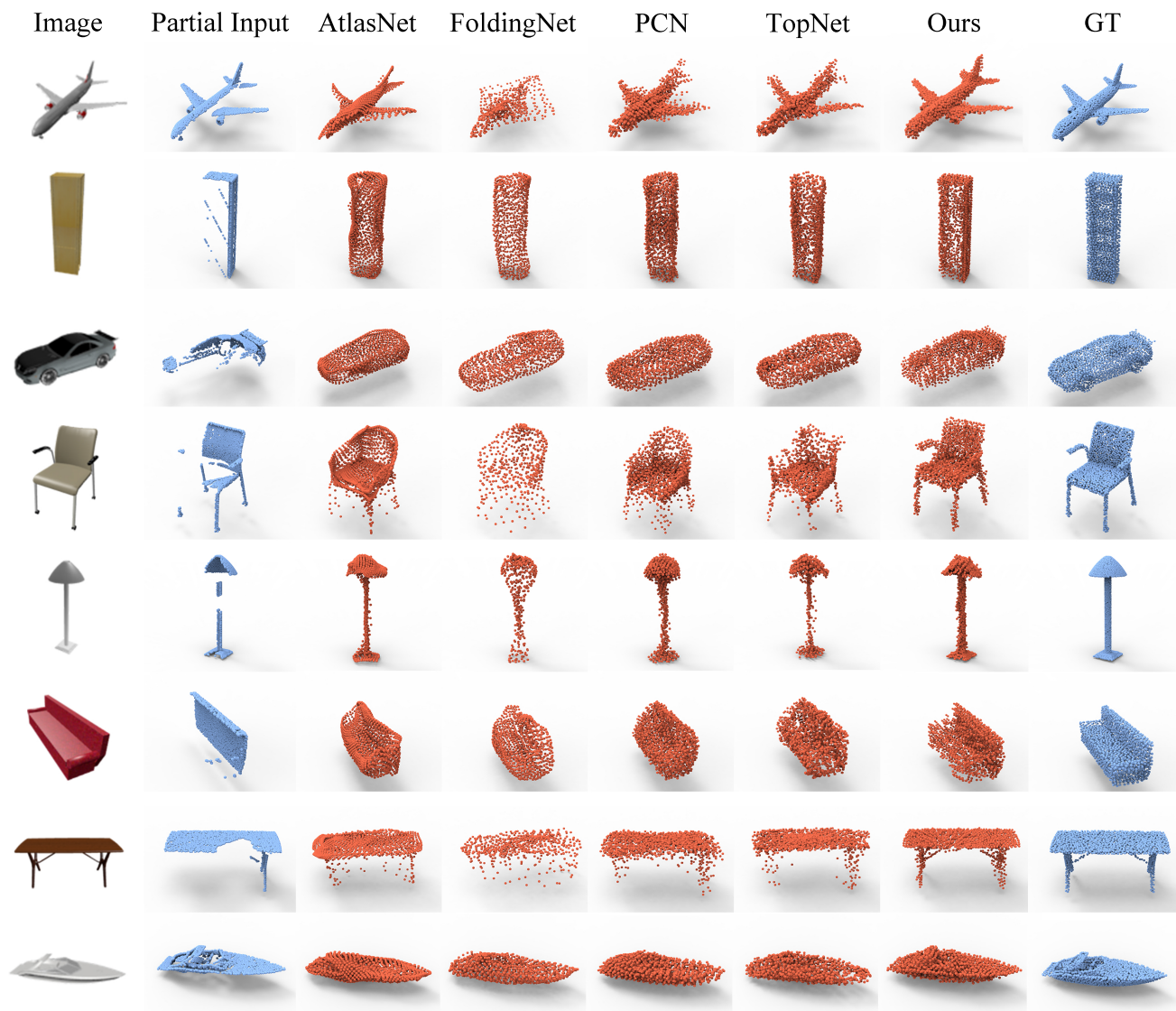
Figure 6. Qualitative comparison on ShapeNet-ViPC. Our method outperforms other baseline methods with significant margins. The resolution for partial, completed and groud truth point clouds are 2,048.


Figure 7. Views provides more complementary information for the input partial point clouds can produce better completion results. Each input partial point clouds are shown on the left; quantitative completion performance measured by the average CD (unit: 10^{-3}) is reported below each input reference view.

Contribution of the Single-view Image. In this set of experiments, we study what kind of input view can better improve the completion. We randomly select $50 \times 8 = 400$ partial point clouds (50 partial point clouds for each category) from the test set of ShapeNet-ViPC for the evaluation. For each partial point cloud, we produce 24 complete point clouds, each of which is generated with the reference of an image from the 24 rendered views. We quantify the completion quality of those 400 completion point clouds with the CD metric and demonstrate some representative results in Figure 7. It indicates that different image views can provide different improvement. The image views which can provide more information for the missing part of the partial point would produce better results.

Category	Mean Chamfer Distance per point			
	P_{rec}	P_{coarse}	P_{global}	$P_{complete}$
Airplane	4.479	2.360	1.993	**1.760**
Cabinet	7.381	5.531	4.807	**4.558**
Car	4.975	3.921	3.308	**3.138**
Chair	12.198	6.967	6.098	**2.476**
Lamp	4.573	3.549	3.186	**2.867**
Sofa	7.809	5.340	4.681	**4.481**
Table	10.967	6.719	5.891	**4.990**
Watercraft	5.626	3.156	2.669	**2.197**
mean	7.241	4.693	4.079	**3.308**

Table 3. Quantitative results for the ablation study; average CD (unit: 10^{-3}) are reported.

Methods	PSG [11]	P_{rec}	$P_{complete}$
CD.(10^{-3})	7.092	7.241	3.308

Table 4. Quantitative comparison with PSG.

Remove	xyz	$f_{partial}$	f_{recon}	f_{pixel}	f_{grid}	none
CD.(10^{-3})	3.348	3.325	3.324	3.321	3.316	3.308
Decline(%)	1.22	0.52	0.49	0.39	0.24	-

Table 5. Quantitative results of removing different features and their relative decline compared with fusion of all features.

Contrast to SVR Methods. In this set of experiments, we study if the proposed ViPC is superior to the SToA single-view based reconstruction method PSG [11]. We compare three different architectures: PSG, the point generation network in the Modality Transfer stage (i.e., P_{rec} in Table 3), and the entire proposed framework (i.e., $P_{complete}$ in Table 3 on the test set of ShapeNet-ViPC. We compute the average CD value of the results produced by the three comparison architectures. We summarize the quantitative results in Table 4 and visualize representative results in Figure 8. Those results indicate the entire proposed framework outperform exceed PSG with a large performance margin. We also find the results generated by PSG show better quality than P_{ren}. Replacing the adopted network in Modality Transfer with a more effective network like PSG may provide a better solution.

Feature Ablation Studies. We remove xyz, $f_{partial}$, f_{recon}, f_{pixel}, f_{grid} from f_{fusion}, respectively, to study the contribution of different features in f_{fusion} during the Part Refinement. The results of completion performance are shown in Table 5. Their contributions can be ranked as $xyz > f_{partial} \approx f_{recon} > f_{pixel} > f_{grid}$.

5.3. Limitations

Point cloud registration. To ensure the Part Filter and Part Refinement giving full play to their effects, the input partial point cloud and the reconstructed point cloud should be aligned accurately. Because the camera parameters are
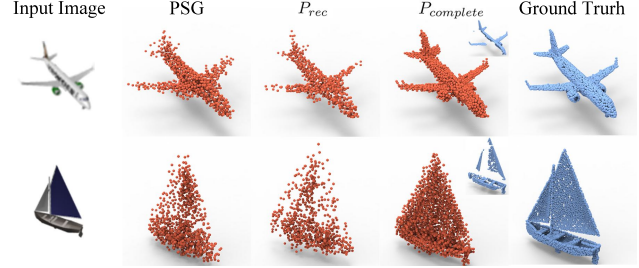

Figure 8. Qualitative comparison with PSG. The input views are shown on the left; extra input partial point clouds for $P_{complete}$ are shown in the top right corner of each corresponding results.

not hard to obtain in calibrated devices. We align the input partial point cloud with the reconstructed point cloud by using camera parameters in Modality Transfer stage. As for non-calibrated devices, we tried several classical unsupervised point cloud registration methods such as ICP [3] and ICP-MCC [10] to perform the point cloud registration. Unfortunately, due to the sparsity of the reconstructed point clouds and the incompleteness of the partial input, it is hard to achieve an accurate alignment. Replacing the current registration solution in Modality Transfer with an effective learning based one is worthy of further study in the future.

Completion in real-world scenes. We have also evaluated the proposed method in real-world scenes where we captured the single view images by mobile phone and collected the partial point clouds by the Li-DAR device in the iPad Pro. Because we train our model on rendered images where the rendered textures can not reflect the illumination in the real environment, it leads to poor-quality reconstructed point clouds in the Modality Transfer stage and thus produces unsatisfactory completion results. Training on real-world data may solve this problem.

6. Conclusion

We propose a pioneer sensor fusion work called ViPC for the task of point cloud completion. ViPC is a view-guided point cloud completion framework. It takes the missing global structure information from an extra single-view image to complete a partial point cloud. The core technical contribution in ViPC is a point cloud refinement network called "Dynamic Offset Predictor" which can deferentially refine the points in a coarse point cloud. We compare ViPC with existing single-modality based STOA methods that reconstruct a complete point cloud based on either the point cloud modality or the image modality. It demonstrates significant quality improvement on a new large-scale dataset we collected for the point cloud completion task.

References

[1] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Cláudio T. Silva. State of the art in surface reconstruction from point clouds. pages 161–185. Eurographics Association, 2014. 2

[2] Dimitri P Bertsekas. A distributed asynchronous relaxation algorithm for the assignment problem. In *IEEE CDC*, pages 1703–1704, 1985. 5

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 8

[4] M Lo Brutto and Paola Meli. Computer vision tools for 3d modelling in archaeology. *International Journal of Heritage in the Digital Era*, 1:1–6, 2012. 1

[5] Edward Castillo, Jian Liang, and Hongkai Zhao. Point cloud segmentation and denoising via constrained nonlinear least squares normal estimates. In *Innovations for Shape Analysis*, pages 283–299. Springer, 2013. 1

[6] 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. *CoRR*, abs/1512.03012, 2015. 2, 5

[7] Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1261–1268, 2010. 2

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Eur. Conf. Comput. Vis.*, pages 628–644. Springer, 2016. 5

[9] Angela Dai, Charles Ruizhongtai Qi, and Matthias Niener. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5868–5877, 2017. 1, 2

[10] Shaoyi Du, Guanglin Xu, Sirui Zhang, Xuetao Zhang, Yue Gao, and Badong Chen. Robust rigid registration algorithm based on pointwise correspondence and correntropy. *Pattern Recognition Letters*, 132:91–98, 2020. 8

[11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 605–613, 2017. 4, 5, 8

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 5, 6

[13] 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 *Int. Conf. Comput. Vis.*, pages 85–93, 2017. 2

[14] W. Hu, Z. Fu, and Z. Guo. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Trans. Image Process.*, 28(8):4087–4100, 2019. 2

[15] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph.*, 31(4):55, 2012. 2

[16] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. pages 287–296, 2018. 1

[17] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29, 2013. 1, 2

[18] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Trans. Graph.*, 32(4):70, 2013. 2

[19] Yi Lin. Lidar: An important tool for next-generation phenotyping technology of high potential for plant phenomics? *Computers and electronics in Agriculture*, 119:61–73, 2015. 1

[20] F. Lozes, A. Elmoataz, and O. Lézoray. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Trans. Image Process.*, 23(9):3896–3909, 2014. 2

[21] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Trans. Graph.*, volume 25, pages 560–568. ACM, 2006. 2

[22] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 4, 5

[23] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. pages 381–389. ACM, 2006. 2

[24] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 1

[25] Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number CONF, pages 23–32, 2005. 2

[26] Mark Pauly, Niloy J Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J Guibas. Discovering structural regularity in 3d geometry. In *ACM Trans. Graph.*, volume 27, page 43. ACM, 2008. 2

[27] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015. 1

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017. 1, 4

[29] Lorenzo Rieg, Volker Wichmann, Martin Rutzinger, Rudolf Sailer, Thomas Geist, and Johann Stötter. Data infrastructure for multitemporal airborne lidar point cloud analysis–

examples from physical geography in high mountain environments. *Computers, Environment and Urban Systems*, 45:137–146, 2014. 1

[30] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2484–2493, 2015. 2

[31] Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. In *Computer Graphics Forum*, volume 28, pages 503–512. Wiley Online Library, 2009. 2

[32] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3405–3414, 2019. 5

[33] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 383–392, 2019. 2, 5, 6

[34] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A field model for repairing 3d shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[35] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Int. Conf. Comput. Vis.*, volume 2, pages 1824–1831, 2005. 1, 2

[36] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Eur. Conf. Comput. Vis.*, pages 52–67, 2018. 4, 5

[37] 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 Trans. Graph.*, 38(5):146:1–146:12, 2019. 1

[38] Yifan Wang, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5958–5967. Computer Vision Foundation / IEEE, 2019. 1

[39] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Eur. Conf. Comput. Vis.*, 2020. 1

[40] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 206–215, 2018. 2, 4, 5, 6

[41] Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, Daniel Cohen-Or, and Baoquan Chen. Morfit: interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. *ACM Trans. Graph.*, 33(6):202–1, 2014. 2

[42] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *International Conference on 3D Vision*, pages 728–737, 2018. 1, 2, 5, 6