Learning Point Cloud Completion without Complete Point Clouds: A Pose-Aware Approach

Jihun Kim, Hyeokjun Kwon, Yunseo Yang, and Kuk-Jin Yoon
Korea Advanced Institute of Science and Technology
{jihun1998,0327june,acorn,kjyoon}@kaist.ac.kr

Abstract

Point cloud completion is to restore complete 3D scenes and objects from incomplete observations or limited sensor data. Existing fully-supervised methods rely on paired datasets of incomplete and complete point clouds, which are labor-intensive to obtain. Unpaired methods have been proposed, but still require a set of complete point clouds as a reference. As a remedy, in this paper, we propose a novel point cloud completion framework without using any complete point cloud at all. Our main idea is to generate multiple incomplete point clouds of various poses and integrate them into a complete point cloud. We train our framework based on cycle consistency, to generate an incomplete point cloud such that 1) shares the same object as the input incomplete point cloud and 2) corresponds to an arbitrarily given pose. In addition, we devise a novel projection method conditioned by pose to gather visible features, from a volumetric feature extracted by an encoder. Extensive experiments demonstrate that the proposed method achieves comparable or better results than existing unpaired methods. Further, we show that our method also can be applied to real incomplete point clouds.

1. Introduction

A point cloud is a commonly used representation of 3D scenes and objects in the fields of computer vision and robotics [14, 20, 21, 22, 28, 9, 24, 2, 3, 7, 15]. However, obtaining complete point clouds is often difficult due to the lack of observation or the limitations of sensors. As a remedy, the point cloud completion task has been spotlighted to restore complete point clouds from incomplete ones.

The existing fully-supervised point cloud completion methods [16, 17, 19, 38, 39, 40, 42, 43, 44] have shown promising results; however, they typically rely on datasets that include incomplete point clouds paired with complete point clouds serving as ground truth (GT). Since obtaining GT complete point clouds is labor-intensive, several works [6, 36, 45, 4] have employed two sets of point clouds: one with incomplete point clouds and the other with complete point clouds, which are not directly paired with each other. This unpaired setting could learn the mapping between the incomplete and complete point clouds, while reducing the reliance on paired GT complete point clouds. Nevertheless, the requirement of a set of complete point clouds still remains a major limitation to the practical application of point cloud completion.

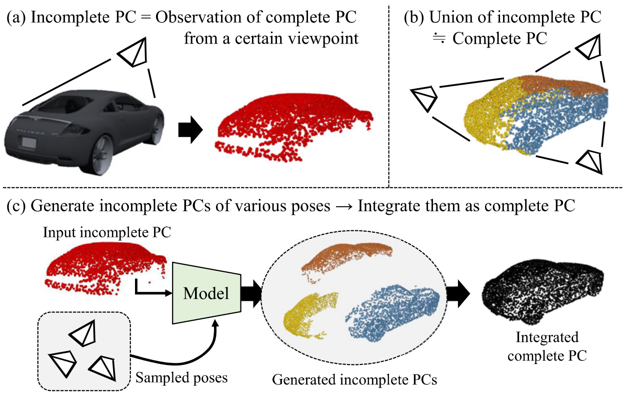
(a) Incomplete PC = Observation of complete PC
from a certain viewpoint
(b) Union of incomplete PC
≒ Complete PC
(c) Generate incomplete PCs of various poses → Integrate them as complete PC
Input incomplete PC
Model
Sampled poses
Generated incomplete PCs
Integrated
complete PC

Figure 1. Illustration of our main idea. We re-interpret the point cloud completion task as generating multiple incomplete point clouds that correspond to various poses and integrating them.

In this paper, we propose a novel point cloud completion framework that **does not use any complete point clouds**. Our main novelty lies in how to learn a mapping from an incomplete point cloud to a complete point cloud when the latter is not available at all. For this, we present a simple yet effective idea shown in Fig. 1. An incomplete point cloud usually results from self-occlusion occurs when observing an object from a certain viewpoint, as in (a). Therefore, if we obtain multiple incomplete point clouds of the object from various viewpoints, a union of them would be equivalent to the complete point cloud of that object. Based on this idea, we formulate the point cloud completion as generating multiple incomplete point clouds using a single input incomplete point cloud and sampled poses. Here, the generated incomplete point clouds should 1) share the same object as the input incomplete point cloud and 2) correspond to the arbitrarily given input poses. If the poses are various enough to cover the entire object, we could obtain the complete point cloud by integrating the incomplete point clouds.

To achieve the above, our framework starts with an encoder that extracts a volumetric shape feature from the input incomplete point cloud. This feature is designed to contain complete information about the shape of the target object. To generate an incomplete point cloud from the volumetric feature, we propose a novel projection method conditioned by the pose, similar to obtaining an incomplete point cloud as a capture of the target object from a certain viewpoint. The method determines the region visible at the given pose and obtains a projected feature by gathering the volumetric feature of that region. A decoder is then trained to generate an incomplete point cloud from the projected feature.

We do not have access to either the complete point cloud or the incomplete point cloud of another pose. Therefore, for training, we propose a novel dataset that provides both incomplete point clouds and corresponding poses, based on the ShapeNet dataset [5]. Note that our dataset includes only one incomplete point cloud per each object, unlike the setting of [13]. Using our dataset, we make our framework reconstruct the input incomplete point cloud itself by gathering the projected feature using the pose corresponding to the input. On the other hand, when using randomly sampled poses, we impose cycle consistency by feeding the generated incomplete point cloud to the encoder/decoder again.

During the inference phase, we extract the volumetric shape feature from the input and obtain the projected features using various poses that cover the entire object. By decoding the features and integrating the generated incomplete point clouds, we can obtain a complete point cloud as the final output. To verify the proposed method, we conduct extensive experiments and comparisons with the existing unpaired point cloud completion methods, on the proposed dataset. The results support that, the proposed method is comparable to the existing studies and sometimes achieves even more substantial completion results, without using any complete point cloud. Further, we conduct an experiment on the KITTI dataset [10] to show that the proposed method can be also applied to the real incomplete point clouds. Our main contributions are as follows:

- We formulate a point cloud completion task as generating and integrating multiple incomplete point clouds corresponding to various poses.
- We propose a novel framework that generates an incomplete point cloud using the input incomplete point cloud and the given arbitrary pose.
- We introduce a new dataset consisting of incomplete point clouds and poses.
- We experimentally verify that the proposed method is comparable or even superior to existing studies, without using complete point cloud at all.

2. Related Works

2.1. Point Cloud Completion

Point cloud completion aims to restore a complete point cloud from an input incomplete point cloud. Although fully-supervised completion methods [8, 17, 19, 23, 32, 34, 37, 40, 43, 46, 35, 31] achieve substantial completion result, they rely on the dataset including paired incomplete point cloud and GT complete point cloud, which is difficult to acquire, especially for real scenes. To overcome the limitation, unpaired approaches [6, 36, 45, 11] have been proposed to learn the mapping between the sets of complete and incomplete point clouds. Pcl2pcl [6] is the first unpaired point cloud completion method, which allows translation between partial and complete latent spaces by adversarial training. Subsequently, Cycle4completion [36] has achieved more advanced results by considering bidirectional geometric correspondence by cycle transformation and missing region coding. ShapeInversion [45] obtained optimal shape code for reconstructing 3D shapes based on the concept of GAN inversion. OptDE [11] proposed cross-domain completion by disentangling shape, domain, and occlusion factors, using domain-invariant geometric information and viewpoint prediction. Although the unpaired methods have shown promising results, their practical application is still limited by the requirement for a number of complete point clouds. [13] has proposed to utilize multiple incomplete point clouds of the same object, obtained from observations at different poses; however, such multiple observations are often not available. Unlike the existing methods, our method does not access the complete point cloud at all, and also uses only one incomplete point cloud per each object, throughout the entire training process.

2.2. Pose-Aware 3D Processing

One of the most representative 3D tasks dealing with the pose is point cloud canonicalization [29, 30, 26]. ConDor [29] takes an un-canonicalized collection of full or partial point clouds obtained through the internet or depth sensor as input. Meanwhile, [33] proposed self-supervised learning via disentangling content and pose information for the point cloud feature representation. As such, numerous studies have been conducted to exploit these two factors, since 3D data usually contains pose attributes as well as the shape of the object. Nevertheless, in the field of point cloud completion, using pose has not been deeply studied despite its usefulness. Since an incomplete point cloud is usually obtained by scanning the object from a single viewpoint through a scanner, information about the viewpoint (*i.e.*, pose) can be beneficial for recovering the self-occluded regions. In this light, we utilize the pose information to complete the point cloud by generating multiple incomplete point clouds.

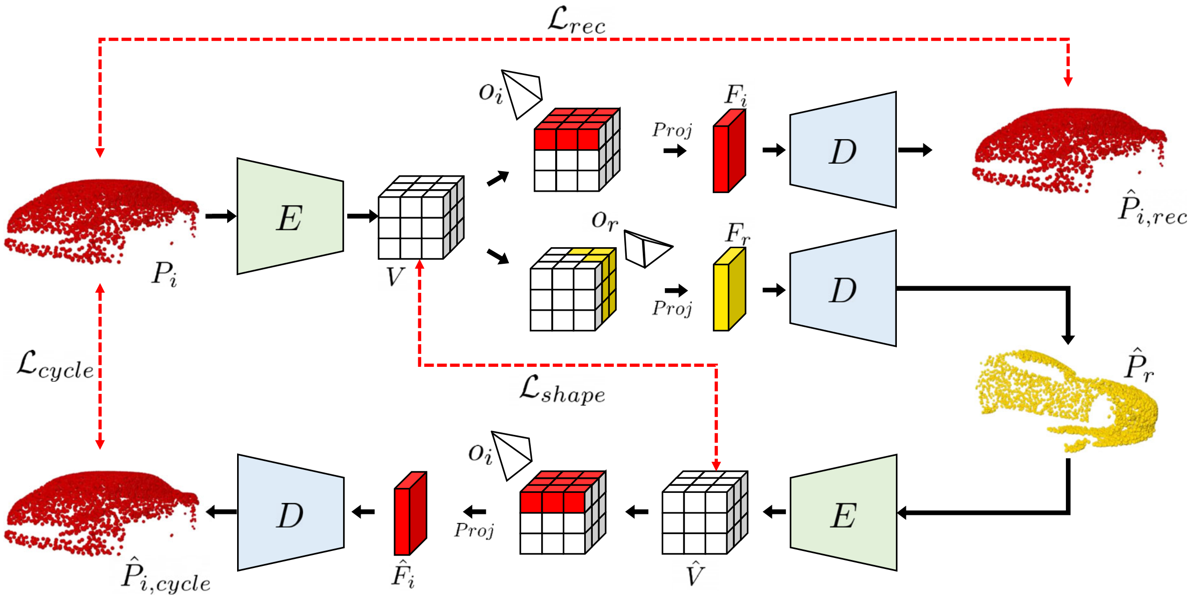

Figure 2. Illustration of a framework for generating a pose-aware point cloud. First, the encoder extracts the volumetric shape feature V from the input point cloud P_i. Then, projected features F_i and F_r are generated by determining the visible region with input pose o_i and random pose o_r. Incomplete point cloud $\hat{P}_{i,rec}$ is decoded to constrain with $\mathcal{L}_{rec}$. Incomplete point cloud $\hat{P}_r$ is decoded and entered into the model again with input pose o_i. After that, volumetric feature $\hat{V}$, projected feature $\hat{F}_i$, and $\hat{P}_{i,cycle}$ are generated sequentially. We use $\mathcal{L}_{shape}$ and $\mathcal{L}_{cycle}$ to constrain our network.

2.3. Volumetric Feature Representation

Volumetric features are commonly used in tasks such as multi-view stereo [25, 27], 3D semantic segmentation [12, 47], and point cloud completion [8, 41, 40] due to their capability to preserve the spatial information of the input. However, standard voxelization can result in geometric information loss and quantization effects. To address this issue, GRNet [40] proposed a differentiable completion network that aggregates features between 3D grid vertices using interpolation. Similarly, our approach also utilizes the volumetric representation and pose to perform completion.

3. Proposed Methods

3.1. Overview

In this paper, we refer to an incomplete point cloud as $P_i \in \mathbb{R}^{N\times 3}$, where N is the number of points in the point cloud, and the subscript i represents the pose at which P_i is captured. For convenience, we define the pose as a two-dimensional vector $o = (\phi, \theta)$, as the incomplete point cloud is usually not significantly affected by the distance between the scanner and the target object.

The goal of our work is to achieve point cloud completion without relying on using complete point clouds. To accomplish this, we generate multiple incomplete point clouds from a single input point cloud, where each generated incomplete point cloud shares the same object as the input but corresponds to a different pose.

As shown in Fig. 2, we begin by using an encoder E to obtain the volumetric shape feature V from the input incomplete point cloud P_i. We assume that the encoder can extract complete shape information of the object from the input incomplete point cloud. In other words, the volumetric shape feature serves as a complete target object at the feature level. To mimic the process of obtaining an incomplete point cloud as a capture of the target object at a certain pose, we gather the features corresponding to the arbitrarily given pose o_r (or o_i for the upper branch) from the volumetric shape feature. Then, with a decoder D, we decode the gathered feature F to generate the incomplete point cloud.

To ensure that the incomplete point cloud generated by our framework correctly corresponds to the given pose, we impose reconstruction loss and cycle consistency loss, which will be explained in Section 3.2. Further, we introduce a novel method to determine the visible region and gather the features on it into the projected feature. We describe the details of the projection method in Section 3.3. Finally, in Section 3.4, we demonstrate how we obtain the complete point cloud using our framework, in the inference phase. It is worth noting that throughout the entire process, we do not use any complete point cloud.

3.2. Pose-Aware Point Cloud Generation

Since we have neither a complete point cloud nor an incomplete point cloud captured from another view, the input incomplete point cloud is the only available data in the point cloud domain that can be used as supervision. Therefore,

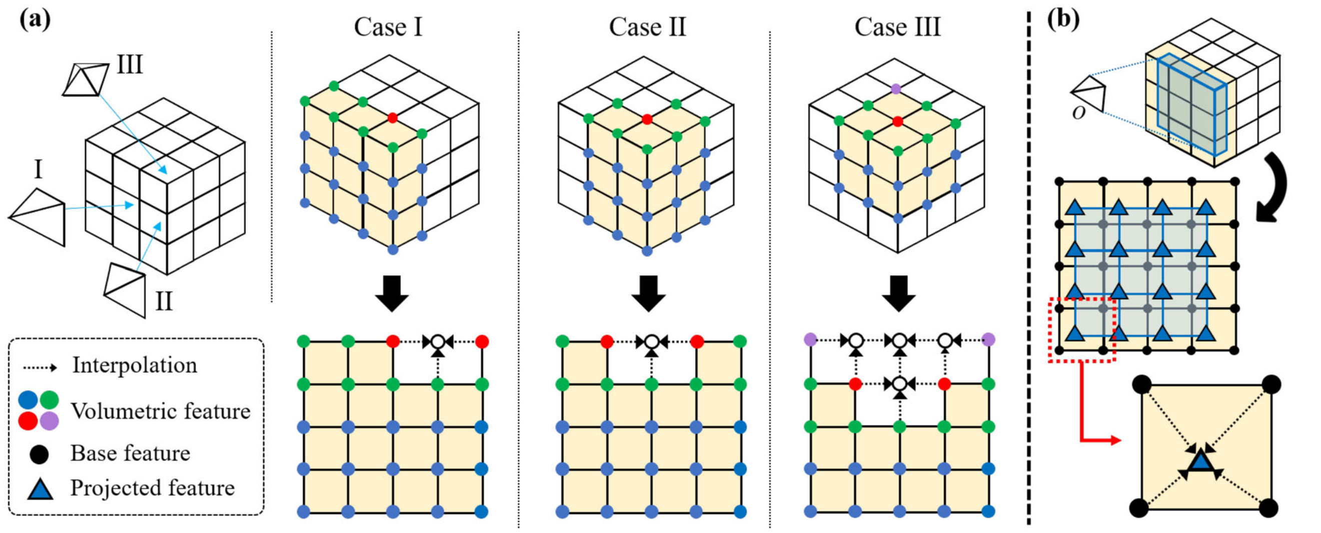

Figure 3. Illustration of our projection method. (a) We first determine the base region (covered by yellow) according to the given pose. Note that any pose can be categorized as one of cases I, II, and III, after a proper rotation. Then, the features on the base region are gathered into a 2D feature grid. (b) The visible region (covered by blue) defined on the grid is obtained by interpolating the neighboring base features.

we devise two branches that can be effectively trained using the input incomplete point cloud. The first branch uses the input pose o_i to generate the projected feature F_i and decode $\hat{P}_{i,rec}$, which should be identical to the input point cloud P_i. To ensure this, we minimize the following reconstruction loss $\mathcal{L}_{rec}$:

$$\mathcal{L}_{rec} = \mathcal{L}_{CD}(P_i\, , \hat{P}_{i,rec}). \tag{1}$$

Here, $\mathcal{L}_{CD}$ denotes the Chamfer Distance (CD), commonly used in the point cloud field. Note that, for two given point clouds P_1 and P_2, the CD is formulated as

$$\mathcal{L}_{CD}(P_1, P_2) = \sum_{p_i \in P_1} \min_{p_j \in P_2} ||p_i - p_j|| + \sum_{p_j \in P_2} \min_{p_i \in P_1} ||p_j - p_i||. \tag{2}$$

Further, we randomly sample an arbitrary pose o_r, which is different from the input pose o_i, and generate projected feature F_r. If we decode F_r, the resulting incomplete point cloud $\hat{P}_r$ should share the same shape with P_i but correspond to the sampled pose o_r. However, as mentioned, we cannot access the GT P_r. Therefore, we impose cycle consistency on our framework. In specific, we put $\hat{P}_r$ into the encoder again and get $\hat{V}$, which is the shape feature of the $\hat{P}_r$. Since the original input P_i and $\hat{P}_r$ should have the same shape, we minimize a shape loss $\mathcal{L}_{shape}$ defined as follows:

$$\mathcal{L}_{shape} = ||V - \hat{V}||_1. \tag{3}$$

Besides, using the $\hat{V}$ and the input pose o_i, we gather the projected feature $\hat{F}_i$ and decode it into $\hat{P}_{i,cycle}$, which should be identical to the input point cloud. Therefore, we define cycle loss $\mathcal{L}_{cycle}$ as follow:

$$\mathcal{L}_{cycle} = \mathcal{L}_{CD}(P_i,\, \hat{P}_{i,cycle}). \tag{4}$$

These loss terms, $\mathcal{L}_{shape}$ and $\mathcal{L}_{cycle}$, enable the encoder to extract the shape feature containing complete shape information and also help the decoder to generate incomplete point cloud that reflects the pose.

To sum up, we train our framework by using a combination of $\mathcal{L}_{rec}$, $\mathcal{L}_{cycle}$, and $\mathcal{L}_{shape}$. The total loss function is defined as follow:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \mathcal{L}_{cycle} + \lambda \mathcal{L}_{shape} \tag{5}$$

where λ is the weight for $\mathcal{L}_{shape}$.

3.3. The Proposed Projection Method

In order to effectively incorporate pose information into our learning framework, we devise the encoder to extract the shape information as a form of volumetric feature, denoted as $V = \{v_i\}_{i=1}^{N^3}$. Here, $v_i \in \mathbb{R}^d$, where N is the resolution of the volumetric feature and d is the feature dimension. Similar to the target object being captured as an incomplete point cloud from a certain viewpoint, we devise a gathering process that selects the features that are visible from an arbitrarily given pose.

However, the volumetric feature is a spatially discrete 3D grid, and thereby a naive visibility test returns the same results for the wide range of poses. To mitigate this, we establish a projection rule that can be applied for arbitrary poses, as shown in Fig. 3 (a). In specific, we subdivide the entire set of poses into the pose subsets, per 30 degrees. Then, we predefine the base region (covered by yellow) for each pose subset. Note that, without loss of generality, any arbitrary pose segment can be categorized into one of the three cases (I, II, and III), after a proper rotation.

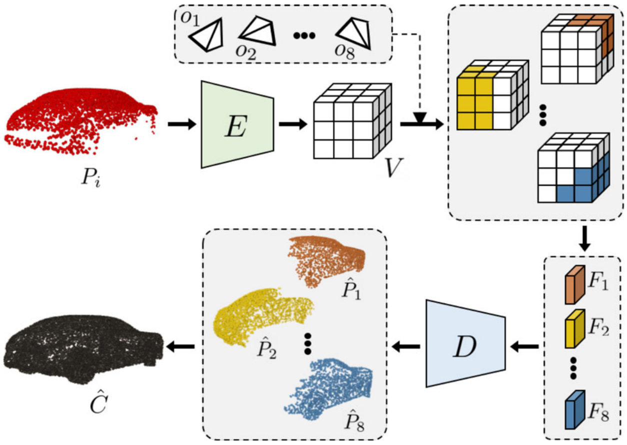

Figure 4. Illustration of obtaining the complete point cloud with the proposed framework in the inference phase. We gather the projected features $F_{1:8}$ from the shape feature V, according to the various poses $o_{1:8}$. Then, each projected feature is decoded into the incomplete point cloud $\hat{P}_{1:8}$. A union of the generated incomplete point clouds serves as the resulting complete point cloud $\hat{C}$.

Once the base region is determined, we unfold the volumetric features located in that region into the 2D grid. During the unfolding, we compute the feature of the empty nodes by interpolating the neighboring volumetric features. In detail, we simply average the neighboring features and use the result as the feature of the node. Then, we define the visible region (covered by blue) on the 2D base grid, as shown in Fig. 3 (b). Here, as θ and ϕ increase, the visible region would move up and right, respectively. Finally, to obtain the feature map that represents the visible region, we perform bilinear interpolation using the neighboring base features. The weights for interpolation are determined according to the values of θ and ϕ. Further details of the projection method are provided in *Supp. Material*.

3.4. Obtaining Complete Point Cloud

As aforementioned, in the inference phase, we generate multiple incomplete point clouds of various poses and integrate them into a complete one. In detail, we first extract the volumetric shape feature V from the input incomplete point cloud, as shown in Fig. 4. From the shape feature, we obtain the projected features $F_{1:8}$ by using a set of poses $o_{1:8}$. Here, we use a predefined set of eight poses, $\{(\frac{\pi}{4}, \frac{\pi}{4})$, $(\frac{\pi}{4}, \frac{3\pi}{4})$, $(\frac{\pi}{4}, \frac{\pi 5}{4})$, $(\frac{\pi}{4}, \frac{7\pi}{4})$, $(\frac{3\pi}{4}, \frac{\pi}{4})$, $(\frac{3\pi}{4}, \frac{3\pi}{4})$, $(\frac{3\pi}{4}, \frac{5\pi}{4})$, $(\frac{3\pi}{4}, \frac{7\pi}{4})\}$, which can cover the entire object. Then, we decode the projected features into the incomplete point cloud $\hat{P}_{1:8}$. Finally, a union of the generated incomplete point clouds serves as the resulting complete point cloud $\hat{C}$. Note that we use the incomplete point cloud as the only input for the inference, and the pose of the input is not required.

We observe that the resulting integrated point cloud $\hat{C}$ has substantial quality. However, our generate-and-integrate approach involves repeated decoding to generate multiple incomplete point clouds, which can be computationally inefficient. In Section 4.3, we conduct an experiment to verify the completion performance while changing the number of used poses. It shows that there is a trade-off between the performance and the number of poses. To alleviate this issue, we propose a strategy named Re-Training (RT). As visualized in Fig. 5, this strategy uses the integrated complete point cloud as supervision for training another completion network.

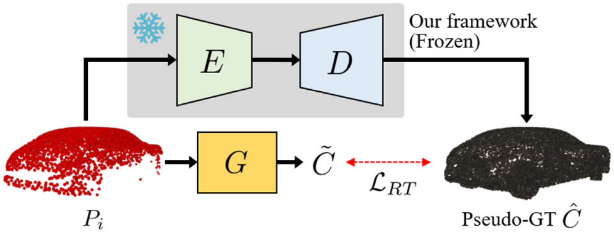

Figure 5. Illustration of the proposed re-training strategy. We use the **frozen** framework to obtain pseudo-GT $\hat{C}$. Then, a new point cloud completion network G is trained to learn a mapping between the input P_i and the pseudo-GT.

First, our framework is trained as in Section 3 and frozen. Then, we obtain an integrated complete point cloud for each incomplete point cloud in the training dataset. Since the obtained complete point cloud is a reasonable completion result, it can serve as a pseudo-GT for the corresponding incomplete point cloud (which was originally used as input). In other words, starting from the initial dataset including incomplete point clouds without any complete point cloud at all, we now have a paired dataset including the incomplete point cloud and the pseudo-GT pairs. Using the incomplete point cloud and pseudo-GT pairs, we train another point cloud completion network G, like the conventional fully-supervised point cloud completion methods. For this, we use the Chamfer Distance loss between the input P_i and pseudo-GT $\hat{C}$, which is denoted as $\mathcal{L}_{RT}$. Note that, we still do not access the real GT complete point clouds at all, throughout the whole process of the re-training.

Compared to directly using the output of our framework as a final output, the re-training can be advantageous in several aspects. First, as we mentioned, re-training is much more efficient and practical in the inference phase, since it requires only one inference per input and can perform multi-class point cloud completion. Further, in terms of the completion performance, the re-training enables learning more complex and beneficial concepts. For example, during re-training, we can expect the network to learn the geometric relation between the incomplete and (pseudo-)complete point clouds. It is especially helpful when the input is severely self-occluded (*e.g.*, when only the upper side of the table can be observed), which cannot be dealt with directly using the pseudo-complete point cloud. In addition, we observe that the re-training can reject some outlier points in the pseudo-GT.

Table 1. Completion results on our dataset. The numbers shown are [CD ↓ / F1 ↑], where CD is Chamfer Distance scaled with $\times 10^4$. Best results are indicated as **bold**.

Methods	Using Complete PC.	Airplane	Cabinet	Car	Chair	Sofa	Table	Vessel	Average
Cycle. [36]	Yes	14.71/81.53	16.64/74.94	35.02/49.33	22.23/70.53	20.38/47.07	21.45/77.18	16.09/76.49	20.93/68.15
ShapeInv. [45]	Yes	**4.16/96.05**	**14.20**/79.91	10.69/87.09	16.67/**81.39**	16.01/**80.82**	23.86/**82.93**	**10.55/ 87.72**	13.74/**85.13**
Ours	No	4.84/95.13	15.11/77.52	9.63/88.01	18.25/75.34	17.50/74.62	19.02/80.93	12.45/84.43	13.73/82.46
Ours (+RT)	No	4.87/94.85	15.14/**81.20**	**9.06/88.84**	**16.22**/77.92	**15.41**/77.67	**18.04**/80.73	10.95/86.36	**12.81**/83.94

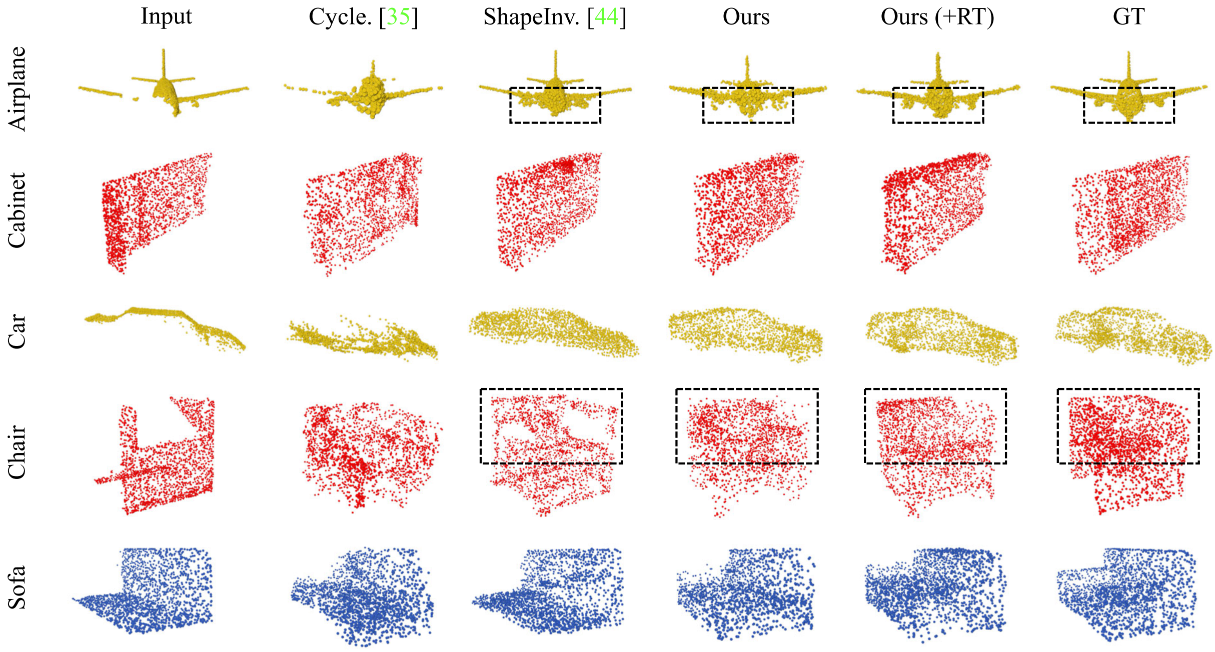

Figure 6. Qualitative results of the point cloud completion methods on our dataset. From left to right: incomplete input, completion results of Cycle4completion [36], ShapeInversion [45], ours, ours (+RT), and GT.

4. Experimental Results

4.1. Dataset Generation

The datasets [8, 34] widely used for the existing point cloud completion studies consist of input point clouds and complete point clouds, which are suitable for the fully-supervised or unpaired settings. On the other hand, to train and evaluate our study, a dataset consisting of complete/incomplete point clouds and corresponding poses is required. Therefore, we build and publish a new dataset for this setting. We use the complete point clouds in the ShapeNet Benchmark [5]. First, we randomly sample the pose as a viewpoint located on the sphere. Then, with the sampled pose, we use [18] to remove unobserved points from the complete point cloud. The resulting point cloud can serve as an incomplete point cloud, where the sampled pose is the corresponding pose. Note that, in our dataset, there is only one incomplete point cloud per the complete point cloud. It means that any complete point cloud cannot be observed from more than one view. More details about the proposed dataset can be found in the *Supp. Material.*

4.2. Comparisons with Other Methods

In order to evaluate the effectiveness of the proposed method, we conduct a comparison with existing methods (**Cycle**. [36] and **ShapeInv**. [45]), which access a set of complete point clouds during training. To ensure a fair comparison, we trained and evaluated all methods on the proposed dataset, using official codes provided by the authors. We evaluate and compare our method with and without using Re-Training (RT) strategy.

Quantitative Results We use Chamfer Distance and F1-score to evaluate the performance of point cloud completion. As shown in Table 1, we evaluate seven categories. Overall, the re-training process improve the performance in most categories, indicating its effectiveness in enhancing the results. Compared to ShapeInv., our proposed method has a lower average Chamfer Distance and a slightly lower F1-score of 1.19. Interestingly, our results outperform Cycle. for all metrics. In other words, our results outperform or achieve comparable performance to existing studies despite not using a complete point cloud.

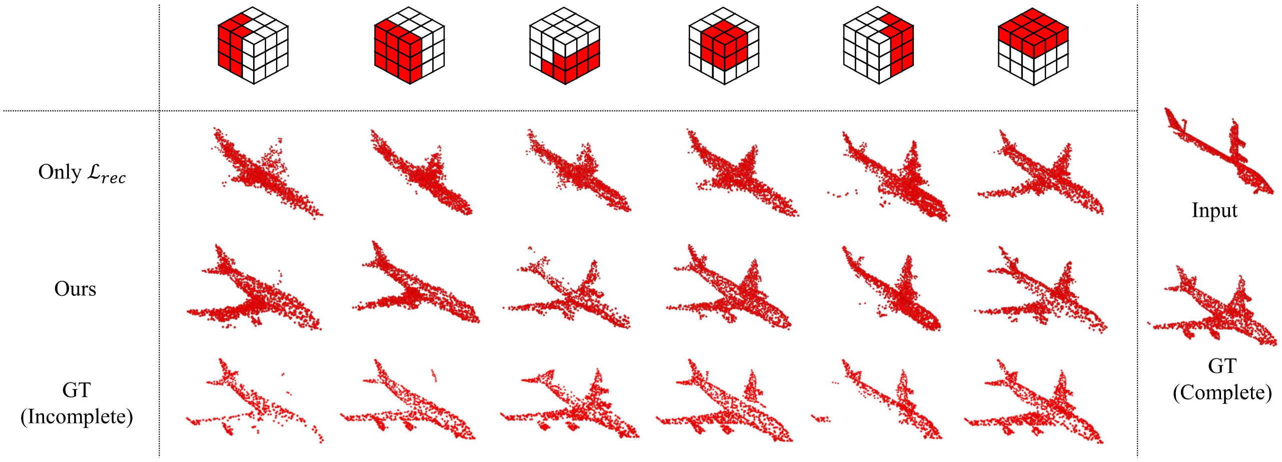

Figure 7. Visualization of pose-aware incomplete point clouds. The cuboids in the top part indicate the volumetric shape features viewed from an arbitrary pose, highlighted in red. The right side of the figure shows the input and ground truth complete point clouds. Two scenarios are considered for generating incomplete point clouds: one that uses only the $\mathcal{L}_{rec}$ loss term, and the other with the full loss term. We also generate ground truth incomplete point clouds to evaluate how well the incomplete point clouds are generated.

Qualitative Results We qualitatively compare the point cloud completion results on our dataset. Figure 6 shows complete point cloud results from existing methods, Cycle., ShapeInv., ours, and ours with re-training technique. Compared to the results of the existing methods, the results support that our method successfully completes the parts that were originally invisible at the input point cloud. Also, we find that the conventional methods fail to restore certain details that are present in the input point cloud. For instance, as shown in the dashed boxes, some fine details such as the exhaust port of the airplane or the armrest parts of the chair are not completely generated, even though they are present in the input point cloud. However, our method can reconstruct them sharply, resulting in a more accurate and realistic airplane model. To sum up, both the quantitative and qualitative results support that the proposed method can achieve substantial completion results, considering that GT complete point cloud is not used at all.

4.3. Ablation Studies

Effectiveness of Each Loss Term We conduct the ablation studies to verify the effectiveness of each term in Eq. 5. As shown in Table 2, we use Chamfer Distance and F1-score as a metric in several categories. The case with only reconstruction loss is used as the baseline. As shown in first row and second row, Chamfer Distance decreases 1.01 CD and F1-score increases 1.30 in average after adding shape loss. We can see that there are performance improvements in every category. As shown in first row and third row, Chamfer Distance decreases 1.55 CD and F1-score increases 1.75 after adding cycle loss. Also, there are performance improvements in every categories. As shown in last row, it shows best performance in both Chamfer Distance and F1-score with all constraints. From these results, proposed constraints are all effective in point cloud completion.

Table 2. Ablation studies on the proposed dataset. The numbers shown are [CD ↓ / F1 ↑], where CD is Chamfer Distance scaled with $\times 10^4$. Best results are indicated as **bold**.

$\mathcal{L}_{rec}$	$\mathcal{L}_{shape}$	$\mathcal{L}_{cycle}$	Airplane	Cabinet	Sofa	Average
✓			5.63/94.11	17.66/73.41	19.97/72.47	14.42/79.99
✓	✓		5.58/94.29	15.86/76.39	18.80/73.19	13.41/81.29
✓		✓	5.08/94.95	15.47/76.38	18.08/73.89	12.87/81.74
✓	✓	✓	**4.84/95.13**	**14.90/88.01**	**17.50/74.62**	**12.42/82.69**

Effectiveness of Number of Poses We examine the effect of the number of input poses on the quality of the complete point cloud generated from our method without re-training. Intuitively, if we use more poses, the generated point clouds that are aware of the input poses and incomplete could cover a wider range of partial views, and thereby complete point cloud would be better. We conduct an experiment with 2, 4, 8, 10, and 14 poses for some categories, where the configurations of 2, 4, and 8 poses are shown in Fig. 8. For a configuration of 10 poses, upward and downward poses are added to the 8 poses. Likewise, additional 4 sideward poses are incorporated into the case of 10 poses for the configuration of 14 poses. Fig. 9 shows how the completion performance changes depending on the number of poses used. As we expected, Chamfer Distance decreases meaningfully as more poses are used until 8 poses. However, the performance becomes saturated as we increase the number of poses to 10 and 14. Therefore, we use 8 poses for our default setting.

4.4. Additional Experiments

Results of Pose-Aware Incomplete Point Cloud Since our main idea is generating and integrating the multiple incom-

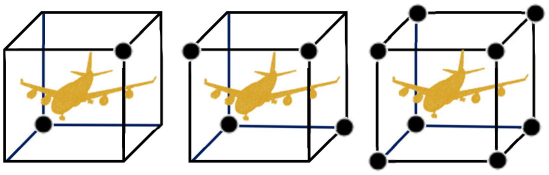

Figure 8. Illustration of pose configurations, which are used to verify the effectiveness of the number of poses. The black dots represent the location of the poses.

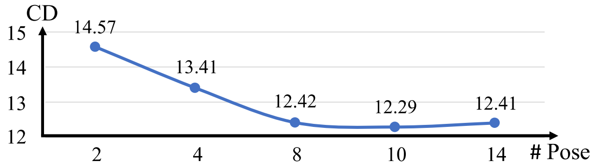

Figure 9. Impact of the number of poses on performance. CD is Chamfer Distance scaled with $\times 10^4$.

plete point clouds corresponding to the given poses, we qualitatively verify whether our network can generate high-quality incomplete point clouds that correspond to the desired poses. Figure 7 shows incomplete point clouds generated from our method. The top row of the figure indicates the various poses we used, and the right side shows the input incomplete input point cloud and the GT complete point cloud. According to them, we generate incomplete point clouds using the models, which are (1) trained with reconstruction loss only and (2) trained with all loss terms. We also provide the GT incomplete point clouds obtained by applying hidden point removal [18] on the GT complete point cloud. When using only $\mathcal{L}_{rec}$, we can observe that the generated incomplete point clouds do not reflect the given poses correctly, and also fail to preserve fine details overall. On the other hand, we can see that incomplete point clouds obtained by the proposed method are well-corresponded to their respective poses appearing as if they had been captured from those viewpoints. Notably, the fine details (*e.g.*, exhaust ports on the wings) are well-completed for all incomplete point clouds, indicating that the network could effectively capture detailed shapes from the input point cloud across various poses. The results support that our framework correctly learns to generate incomplete point clouds according to the given poses, and achieve substantial completion results in the ultimate.

Table 3. Effect of noise in pose. The numbers shown are [CD ↓ / F1 ↑], where CD is Chamfer Distance scaled with $\times 10^4$. Best results are indicated as **bold**.

σ	Airplane	Cabinet	Sofa	Average
0.1	4.94/95.22	16.79/75.05	19.79/72.08	13.84/80.79
0.03	4.89/**95.23**	15.44/77.42	18.07/73.38	12.80/82.01
0	**4.84**/95.13	**14.90/78.31**	**17.50/74.62**	**12.42/82.69**

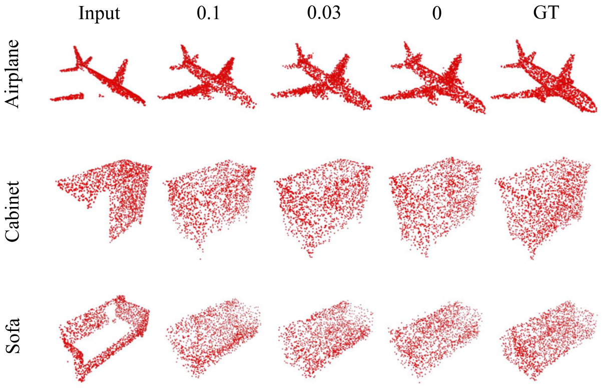

Figure 10. Qualitative results with noisy poses. 0.1, 0.03, and 0 are value of σ in Eq. 6, 7.

Using Noisy Poses In the real scenario, there can be a noise in the poses corresponding to the obtained incomplete point cloud. Therefore, we conduct an experiment to verify the sensitivity of the proposed method against the noise in the poses that we used for training. We observe that our method is robust to the noise in poses to some extent. We conduct an experiment by introducing Gaussian noise to the given poses. Specifically, we add a noise sampled from normal distribution to the given pose $o_i = (\phi, \theta)$. The resulting noisy pose, denoted as ϕ_{noise} and θ_{noise}, is defined as follows:

$$\phi_{noise} = \phi + \sigma N(0,1) \times 180^\circ \tag{6}$$

$$\theta_{noise} = \theta + \sigma N(0,1) \times 180^\circ . \tag{7}$$

Here, $N(0,1)$ is a normal distribution with mean 0 and variance 1. We train and evaluate the model by varying the value of σ. We set the σ as 0.03 and 0.1. For these values, the errors in the pose are mostly within 11° and 36°, respectively. Table 3 demonstrates the impact of the noise on pose information. A value of $\sigma = 0$ means that there is no noise in the pose, which is the default setting provided in the main paper. As noise increases, the average performance decreases compared to the case without any noise. The average Chamfer distance increases by 1.42 CD, and F1-score decreases by 1.90 compared to the case without noise. It should be noted that the effect of noise on performance is not significant. Figure 10 shows the qualitative results when adding noise to the pose. Although the pose is perturbed, the generated complete point clouds remain similar to the case without any noise. From these results, we can conclude that errors in pose have a limited effect on performance.

Completing the Real Incomplete Point Clouds To assess the scalability of our method, we conduct an experiment using the real dataset. We train and test our framework on the point cloud data in KITTI [10] and nuScenes [1], using the provided bounding boxes to calculate the relative pose. Since the real dataset does not include incomplete point clouds observed from upward or downward directions, we enforce a constraint to ensure that the sampled poses fall within a range of the poses of the incomplete point

Table 4. Quantitative comparison on KITTI dataset [10]. Minimum Matching Distance (MMD, ↓) and Unidirectional Chamfer Distance (UCD, ↓) are scaled with $\times 10^3$ and $\times 10^4$, respectively.

	Cycle. [36]	ShapeInv. [45]	Ours
MMD	32.2	**1.6**	2.4
UCD	12.76	3.61	**1.47**

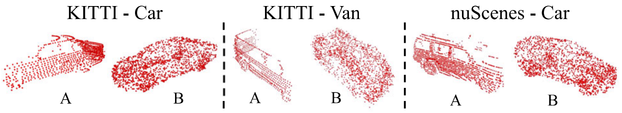

Figure 11. Completion results of Real dataset. (**A**: Real incomplete point cloud, **B**: Our result)

clouds. Table 4 compares the quantitative performance of Cycle.[36] and ShapeInv.[45] using Minimum Matching Distance (MMD) and Unidirectional Chamfer Distance (UCD). To calculate MMD, we use complete point clouds in ShapeNet [5]. MMD evaluates the quality of the result in terms of how realistically the point cloud is generated, and ShapeInv. shows the best performance in this regard. However, our approach also shows comparable performance. Additionally, our approach outperforms the others on the UCD metric, which evaluates how well the generated point cloud reflects the input shape. We show the qualitative results in Fig. 11. In each section, the left side displays input incomplete point clouds, while the right side showcases predicted complete point clouds. Our network successfully generates the complete shape of cars and vans, even when the input point clouds are severely sparse. We can also observe that the obtained complete point clouds well-reflect the detailed shapes. These results demonstrate the potential practicality of our method for real-world applications.

5. Conclusion

In this paper, we propose a novel method for point cloud completion that does not require any complete point clouds. The proposed method generates multiple incomplete point clouds corresponding to various poses and integrates them to obtain the complete point cloud. Our method employs an encoder-decoder framework where the encoder extracts a volumetric shape feature from the input incomplete point cloud, and the decoder generates pose-aware incomplete point clouds. To train the model, we introduce a novel dataset with incomplete point clouds and corresponding poses. The proposed method achieves comparable results with the existing state-of-the-art methods and outperforms some categories. Additionally, we demonstrate that the method can be applied to real incomplete point clouds. In future work, we plan to use a neural implicit representation for the shape feature instead of the volumetric feature, which can achieve better 3D representation capability. In addition, we will further enhance the projection method using a visibility check with learnable opacity.

Acknowledgments This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2022R1A2B5B03002636) and Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by Korea government(MSIT) (No.2020-0-00440, Development of Artificial Intelligence Technology that Continuously Improves Itself as the Situation Changes in the Real World).

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 8

[2] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2021. 1

[3] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10478–10485, 2020. 1

[4] Yingjie Cai, Kwan-Yee Lin, Chao Zhang, Qiang Wang, Xiaogang Wang, and Hongsheng Li. Learning a structured latent space for unsupervised point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5543–5553, 2022. 1

[5] 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. 2, 6, 9

[6] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. In *International Conference on Learning Representations*, 2019. 1, 2

[7] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 1

[8] 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, 3, 6

[9] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 8, 9

[11] Jingyu Gong, Fengqi Liu, Jiachen Xu, Min Wang, Xin Tan, Zhizhong Zhang, Ran Yi, Haichuan Song, Yuan Xie, and Lizhuang Ma. Optimization over disentangled encoding: Unsupervised cross-domain point cloud completion via occlusion factor manipulation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 517–533. Springer, 2022. 2

[12] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 3

[13] Jiayuan Gu, Wei-Chiu Ma, Sivabalan Manivasagam, Wenyuan Zeng, Zihao Wang, Yuwen Xiong, Hao Su, and Raquel Urtasun. Weakly-supervised 3d shape completion in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 283–299. Springer, 2020. 2

[14] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and CL Chen. 3dviewgraph: Learning global features for 3d shapes from a graph of unordered views with attention. *arXiv preprint arXiv:1905.07503*, 2019. 1

[15] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 1

[16] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3d shape completion with multi-view consistent inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10997–11004, 2020. 1

[17] 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. 1, 2

[18] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, pages 24–es. 2007. 6, 8

[19] 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. 1, 2

[20] Xinhai Liu, Zhizhong Han, Fangzhou Hong, Yu-Shen Liu, and Matthias Zwicker. Lrc-net: Learning discriminative features on point clouds by encoding local region contexts. *Computer Aided Geometric Design*, 79:101859, 2020. 1

[21] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Fine-grained 3d shape classification with hierarchical part-view attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2021. 1

[22] Xinhai Liu, Zhizhong Han, Xin Wen, Yu-Shen Liu, and Matthias Zwicker. L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 989–997, 2019. 1

[23] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[25] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020. 3

[26] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7688–7697, 2019. 2

[27] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 3

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[29] Rahul Sajnani, Adrien Poulenard, Jivitesh Jain, Radhika Dua, Leonidas J Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16969–16979, 2022. 2

[30] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24993–25005. Curran Associates, Inc., 2021. 2

[31] 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

[32] 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. 2

[33] Meng-Shiun Tsai, Pei-Ze Chiang, Yi-Hsuan Tsai, and Wei-Chen Chiu. Self-supervised feature learning from partial point clouds via pose disentanglement. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1031–1038. IEEE, 2022. 2

[34] 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, 6

[35] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Learning local displacements for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1568–1577, 2022. 2

[36] 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. 1, 2, 6, 9

[37] 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

[38] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7443–7452, 2021. 1

[39] 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. 1

[40] 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. 1, 2, 3

[41] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. Dense 3d object reconstruction from a single depth view. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):2820–2834, 2018. 3

[42] Kangxue Yin, Hui Huang, Daniel Cohen-Or, and Hao Zhang. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 1

[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

[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. 1

[45] Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1768–1777, 2021. 1, 2, 6, 9

[46] 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

[47] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 3