# Syn-to-Real Domain Adaptation for Point Cloud Completion via Part-based Approach

Yunseo Yang*, Jihun Kim*, and Kuk-Jin Yoon

Visual Intelligence Lab., KAIST
{acorn, jihun1998, kjyoon}@kaist.ac.kr

**Abstract.** Acquiring complete point clouds for real-world scenarios is labor-intensive, making it impractical for conventional learning-based approaches. Numerous methods have been proposed to overcome this limitation by leveraging synthetic complete point clouds. While access to complete point clouds offers a notable advantage, they often struggle to bridge domain gaps, leading to sub-optimal performance. As a remedy, we propose a novel part-based framework for synthetic-to-real domain adaptation in point cloud completion. Our approach starts on the observation that domain gaps inherent in part information are relatively small, as parts are shared properties across categories regardless of domains. To employ part-based approach to point cloud completion, we introduce Part-Based Decomposition (PBD) module to generate part input point clouds. Subsequently, we design a Part-Aware Completion (PAC) module, which operates in a part-wise manner to produce complete point clouds. Within PAC, we devise a novel part-aware transformer to learn relationships between parts and utilize this information to infer missing parts in incomplete point clouds. Extensive experiments demonstrate that our part-based framework significantly outperforms existing studies on real-world point cloud datasets. The code is available at https://github.com/yun-seo/PPCC

**Keywords:** Point Cloud Completion · Synthetic to Real Adaptation

## 1 Introduction

Point cloud completion aims to reconstruct the original shape from the incomplete point cloud. Most learning-based methods [4, 7, 11, 14, 15, 17, 23, 28, 33, 37, 41–43, 45–47, 49–51] are conducted under synthetic data, such as 3D CAD models [2, 40] to learn point cloud completion in a fully supervised manner. Although these studies have shown promising results, there are limitations in application for real-world scans. Direct employment of a completion model trained on synthetic data produces low-quality results due to data distribution discrepancy. Furthermore, the training procedure of the conventional supervised learning method cannot be executed in real-world scans since acquiring complete point clouds corresponding to incomplete point clouds is challenging.

* Equal contribution.

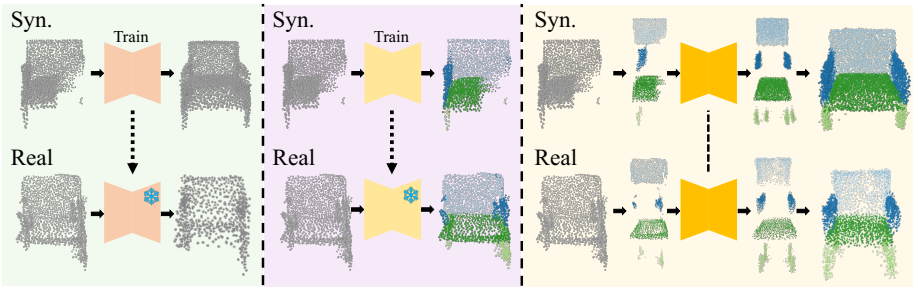


**Fig. 1:** Illustration of our main ideas. (a) We train point cloud completion on the synthetic domain and evaluate them on the real domain. (b) We train part segmentation on the synthetic domain and evaluate them on the real domain. (c) We decompose the point cloud as parts, complete them, and integrate them to generate a complete point cloud both on synthetic and real domains.

Several methods [6, 9, 10, 13] have been proposed to address the above limitations. They alleviate the limitation in real-world scans by imposing self-supervised training procedures or introducing alternative data as supervision instead. Nevertheless, completion results of these approaches still show low quality in some cases since their methods cannot access complete point clouds.

Other studies [3, 8, 12, 19, 20, 36, 38, 39, 48] have proposed the point cloud completion of real-world scans by incorporating synthetic datasets as additional training resources. This approach offers an apparent advantage: it enables the network to access complete point clouds while retaining practical applicability to real-world scans. To address the issue posed by the domain gap, they rely solely on adversarial learning processes by incorporating discriminator-like modules. However, this approach is insufficient in bridging the domain gap between synthetic and real point clouds, leading to sub-optimal performance.

In this light, a crucial question arises: what is the key factor for bridging domain gaps in point clouds? Parts, which represent semantic and geometrical divisions within an object, are shared information across categories regardless of the domain. Therefore, we hypothesize that the domain gap inherent in part information might be relatively smaller than that in point cloud completion. To test this hypothesis, we have conducted a toy experiment. We have trained a network for both point cloud completion and part segmentation on synthetic data and then used these models to perform inference on real data. Remarkably, while part segmentation is consistently performed across both domains, point cloud completion shows inconsistent results, as illustrated in Fig. 1 (a) and (b).

Building on this finding, we propose a novel point cloud completion framework grounded in a **part-based approach** under a synthetic-to-real domain adaptation setting as shown in Fig. 1 (c). Since part information shows domain invariance, we hypothesize that part-wise processing could effectively address the domain gap. Additionally, considering the perspective of completion, miss-

ing regions in incomplete point clouds can be inferred by utilizing relationships between parts. Accordingly, we introduce the Part-Based Decomposition (PBD) module, designed to decompose point clouds for both synthetic and real domains to implement part-wise completion. In the synthetic domain, where part labels are available, point clouds are decomposed into part input sets using these labels. However, in the real domain, where part labels are not provided, we employ a pre-trained part segmentation network to obtain part predictions. Leveraging the findings illustrated in Fig. 1, we consistently decompose input point clouds from the real domain into part input sets based on these part predictions.

Following this, we introduce the Part-Aware Completion (PAC) module, designed to process part input sets and execute completion tasks. PAC starts with an encoder, shared across all part inputs, that extracts part features. Part features from each part input encapsulate essential part information, including geometry and semantics. To foster an understanding of the relationships between parts crucial for completion, we propose a novel part-aware transformer. This component facilitates the acquisition of part-aware features, enabling the model to infer missing parts based on existing parts. A shared decoder, employed across all parts, then generates completed point clouds for each part. Note that these processes are conducted in a part-wise manner, which is advantageous for addressing domain gaps. The resultant part point clouds are integrated to construct a complete point cloud corresponding to the input. Furthermore, we introduce a refinement module to enhance the quality of shapes, especially in regions with boundaries between adjacent parts.

Our main novelty lies in leveraging part information to narrow the domain gap between synthetic and real data to perform point cloud completion on real-world datasets successfully. We validate the effectiveness of our approach through extensive experiments and comparisons with existing methods. Our method achieves state-of-the-art (SOTA) results on USSPA dataset [19], surpassing the previous point cloud completion methods. Furthermore, we conduct experiments without part labels in synthetic datasets and still achieve the best performance, which demonstrates the practicality of our methods.

In summary, our key contributions are as follows:

- We find that exploiting part information is beneficial for bridging the domain gaps in point cloud completion task.
- We propose a novel part-based framework for synthetic-to-real domain adaptation in point cloud completion.
- We extensively conduct experiments on real-world datasets to demonstrate the superiority of our method over existing approaches.

## 2 Related Works

**Point cloud completion** Point cloud completion has been studied for the purpose of reconstructing complete point clouds from incomplete point clouds. Starting with PCN [47], various architectures such as folding-based [17,25], voxel-based [5, 32, 43], and transformer-based [4, 15, 42, 46, 50] show promising results,

conducted in a fully supervised manner. Nevertheless, directly utilizing the above methods on real-world datasets leads to low quality on complete point clouds. In addition to that, we cannot apply supervised methods on real-world datasets since complete point clouds do not exist in most cases. Recognizing the limitations of a fully supervised setting, several methods have emerged. For instance, [3, 12, 36, 38, 48] propose methods in an unpaired setting, where incomplete point clouds and unpaired complete point clouds of the same category are used. Typically, these approaches leverage pre-trained networks on complete point cloud sets. However, their effectiveness diminishes when confronted with unconventional shapes, resulting in limited performance.

**Point cloud completion in real-world scans** In most cases, synthetic point cloud datasets consist of complete point clouds, while real-world datasets typically include only incomplete point clouds. In response to this, several studies [6, 9, 10, 13] employ incomplete point clouds only as input and supervision for self-training or introducing alternative data as supervision instead. Despite these efforts, a prevailing limitation that detailed shapes are not generated remains. To address this issue, some studies that leverage the advantage of accessing both the distribution of incomplete and complete point clouds have been proposed. OptDE [8] introduces an architecture that utilizes viewpoints as a key factor to bridge synthetic and real-world point clouds. UGAAN [20] combines an autoencoding network with a GAN architecture and extracts clean shape features by eliminating noise from real-world scans through a discriminator. SCoDA [39] proposed a cross-domain feature fusion module that combines global and local features, and volume consistency self-training that adjusts viewpoints to produce incompleteness to different extents. USSPA [19] proposes an architecture that capitalizes on the inherent symmetry found in most objects in real-world scans. Even though many works have achieved substantial improvements on real-world datasets, they still have issues bridging domain gaps, which hinders point cloud completion in real-world point clouds.

**Part-based shape analysis** Objects in the same category can be decomposed into parts with similar semantics and geometries. Taking advantage of these attributes, part-based shape analysis has been extensively explored in various fields such as part retrieval, shape abstraction, generation, and reconstruction [16, 22, 29, 29, 31, 44]. The utilization of part information facilitated detailed shape manipulation by leveraging relationships among individual components, thereby enhancing the overall performance of the previous research. Most studies learn part representation using densely annotated labels [21] to build part latent space. Even without part labels, it is possible to obtain part-level spatial relationships through research such as co-segmentation [24, 27] and primitive estimation [18, 26]. Additionally, some studies [1, 35] mitigate domain shifts by exploiting the part representation that is more likely to be shared across distinct domains, thereby enhancing generalizability to variations in geometry. Building upon these research trends, we propose the part-based point cloud completion framework to narrow the domain gap between synthetic and real-world datasets.

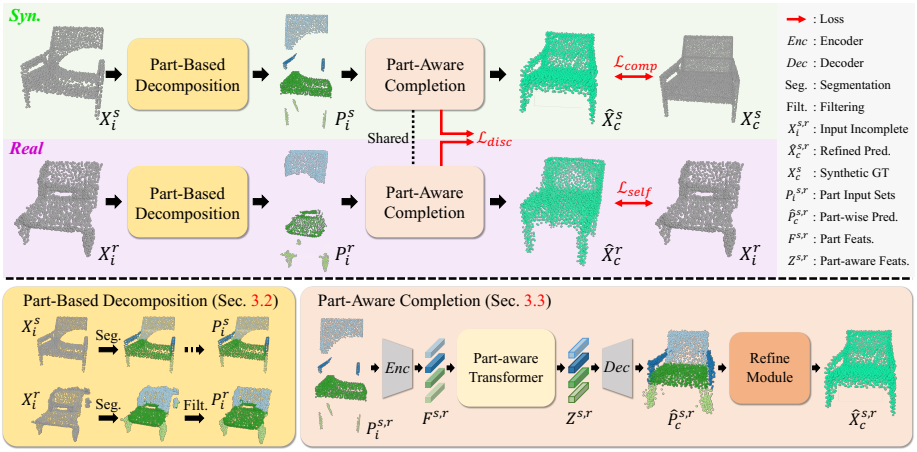


**Fig. 2:** Visualization of our part-based point cloud completion framework. Our framework is composed of Part-Based Decomposition (PBD) module and Part-Aware Completion (PAC) module. Initially, incomplete point clouds from both synthetic and real domains ($X_i^{s,r}$) are decomposed into part input sets ($P_i^{s,r}$) via the PBD. These part input sets are then processed by the PAC, which is shared across both synthetic and real datasets, to generate complete point clouds. The process involves an encoder to obtain part feature sets ($F_i^{s,r}$), a part-aware transformer to learn relationships between parts and get part-aware features ($Z_i^{s,r}$), and a decoder to produce completed part point clouds. These completed part point clouds are integrated to form one complete point cloud ($\hat{P}_c^{s,r}$). Additionally, a refinement module is introduced to improve $\hat{P}_c^{s,r}$, resulting in the final results ($\hat{X}_c^{s,r}$). For optimization, we use $\mathcal{L}_{comp}$, $\mathcal{L}_{self}$ and $\mathcal{L}_{disc}$.

## 3 Method

### 3.1 Overview

In this paper, we represent a point cloud as $X \in \mathbb{R}^{N\times 3}$ where $N$ denotes the number of points. Part labels are represented as $L = \{l_1, ..., l_N\}$ where $l_i \in \{1, ..., n\}$ and $n$ is the number of parts. Accordingly, we define the synthetic domain set as $D^s = \{X_i^s, X_c^s, L^s\}$, where each represents the incomplete point cloud, complete point cloud, and semantic label of the synthetic dataset, respectively. Similarly, we define the real domain set as $D^r = \{X_i^r\}$, where $X_i^r$ represents the incomplete point cloud of the real dataset. Our objective is to achieve point cloud completion in $D^r$, where no complete point clouds are given, by utilizing both $D^s$ and $D^r$ in the training step.

As shown in Fig. 2, our method begins by inputting incomplete point clouds $X_i^s$ and $X_i^r$ into a Part-Based Decomposition (PBD) module. Part input sets of $n$ parts, $P_i^s = \{p_k^s\}_{k=1}^n$ and $P_i^r = \{p_k^r\}_{k=1}^n$, are obtained from the PBD which will be explained in Sec. 3.2. Subsequently, the obtained part input sets $P_i^s$ and $P_i^r$ are entered into the Part-Aware Completion (PAC) module to generate the resultant complete point clouds $\hat{X}_c^s$ and $\hat{X}_c^r$, which will be explained in

Sec. 3.3. Here, networks in the PAC module are shared for both synthetic and real domains. It is noteworthy that our method operates without utilizing any complete point clouds from $D^r$, which is the main objective domain.

### 3.2 Part-Based Decomposition

We start with decomposing the input point clouds to implement a part-wise completion procedure. To decompose input point clouds, the segmentation (Seg.) process should be accomplished first. Part segment labels are provided for incomplete point clouds in the synthetic domain, enabling decomposition based on these labels. In contrast, point clouds from the real domain lack part segment labels, necessitating the incorporation of other methods. To mitigate this, we employ a pre-trained part segmentation network trained on synthetic datasets for the acquisition of real-world part labels. Since the domain gap in part segmentation is low, obtained real-world part segmentation predictions are consistent with part labels in synthetic domain. It is worth noting that this process does not require any training or fine-tuning.

Real-world point clouds often contain noisy points due to the acquisition process or sensor limitations. These points are ambiguous for assigning part segments, and utilizing them can cause negative effects on afterward procedures. Therefore, we define a margin-based reliability metric to filter out noisy points. This metric is measured by finding the maximum and second maximum values of the segmentation logits and calculating the difference between them. Noisy and ambiguous points can be deleted from the filtering (Filt.) process of points with low $r_i$. The filtering process is implemented by removing a certain proportion of points. Through the above processes, part-based decomposed point clouds are acquired for both domains. Details are explained in Sec. 1.1 of *Supp.*

### 3.3 Part-Aware Completion

To obtain part information such as geometry or semantics from part sets, we begin by using an encoder $Enc$, shared among part input sets to extract part feature sets for $n$ parts, denoted as $F^{s,r} = \{f_k^{s,r}\}_{k=1}^n$. For parts not existing in the input point cloud, we represent their part features using a zero vector since part features cannot be extracted for these unseen parts.

The extracted part feature sets are fed into a part-aware transformer to generate part-aware feature sets $Z^{s,r} = \{z_k^{s,r}\}_{k=1}^n$, which encompass both the part information itself and the relationships between each part. To ensure the part-awareness of our transformer, we introduce learnable part tokens for the $n$ parts, denoted as $T = \{t_k\}_{k=1}^n$. By leveraging these part tokens, it also becomes feasible to deduce information about parts that were absent in the input point cloud from other parts. We employ a multi-head self-attention block as the transformer block, with query $Q$, key $K$, and value $V$ defined as:

$$Q = W_Q[\{f_k^s + t_k\}_{k=1}^n]^T,\ K = W_K[\{f_k^s + t_k\}_{k=1}^n]^T,\ V = W_V[\{f_k^s + t_k\}_{k=1}^n]^T, \quad (1)$$

where $W_Q, W_K, W_V$ are learnable matrices. Utilizing these terms, attention $A$ is calculated using the scaled-dot attention metric as:

$$A = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{2}$$

where $d_k$ is a scale factor, and the part-aware feature sets are computed using a feed-forward network composed with MLPs.

To generate complete point clouds, we put the part-aware features into a shared decoder, denoted as *Dec*. This decoder generates $n$ individual part point clouds, $\{\hat{p}_k^{s,r}\}_{k=1}^n$ from the part-aware features. Then, we integrate these part point clouds to produce the complete point clouds, denoted as $\hat{P}_c^{s,r}$. Through the part-wise decoding process, each part point cloud contains more detailed shape compared to decoding the entire shape in a single step. However, one drawback is that the boundaries between adjacent parts are not smoothly generated solely by generating and integrating each part point cloud. To address this issue, we introduce a refinement module composed of MLPs to improve $\hat{P}_c^{s,r}$, resulting in a final refined prediction, $\hat{X}_c^{s,r}$. Details are shown in Sec. 1.2 of *Supp.*

### 3.4 Optimization for Domain Adaptive Point Cloud Completion

Our goal is to perform point cloud completion in real domain successfully by leveraging the merit of synthetic domain that both complete and incomplete point clouds exist. To accomplish this, incomplete point clouds from both synthetic and real domain are utilized with three loss functions during training.

First of all, the network needs to learn how to generate a complete point cloud from an incomplete one. In this regard, we define the completion loss $\mathcal{L}_{comp}$ in the synthetic domain as follows:

$$\mathcal{L}_{comp} = \lambda_1 \sum_{k=1}^{n} \mathcal{L}_{CD}(\hat{p}_k^s, p_k^*) + \lambda_2 \mathcal{L}_{CD}(\hat{X}_c^s, X_c^s) \tag{3}$$

where $\lambda_1$ and $\lambda_2$ are weight parameters for each term, $n$ is number of parts and $p_k^*$ denotes the ground truth (GT) part point cloud generated from $X_c^s$ and $L^s$. Here, $\mathcal{L}_{CD}$ denotes the Chamfer Distance (CD), formulated as:

$$\mathcal{L}_{CD}(P_1, P_2) = \frac{1}{|P_1|} \sum_{\mathbf{x} \in P_1} \min_{\mathbf{y} \in P_2} \|\mathbf{x} - \mathbf{y}\| + \frac{1}{|P_2|} \sum_{\mathbf{x} \in P_2} \min_{\mathbf{y} \in P_1} \|\mathbf{x} - \mathbf{y}\| \tag{4}$$

where $P_1$ and $P_2$ refer to any point clouds. As shown in first term of Eq. (3), we apply CD for each part point clouds with GT part point clouds to generate complete parts. Subsequently, we apply CD between the refined complete point cloud and the complete GT to facilitate the learning process of the refinement module.

For real domain, we define the self-supervised loss $\mathcal{L}_{self}$ by utilizing the input real point cloud as follows:

$$\mathcal{L}_{self} = \lambda_3 \mathcal{L}_{UCD}(\hat{X}_c^r, X_i^r) \tag{5}$$

where $\lambda_3$ is weight parameter for the self-supervised loss. In contrast to $\mathcal{L}_{comp}$, applying CD with the input real point clouds leads to generating incomplete point clouds. Therefore, we introduce $\mathcal{L}_{UCD}$, formulated as:

$$\mathcal{L}_{UCD}(P_1, P_2) = \frac{1}{|P_2|} \sum_{\mathbf{x} \in P_2} \min_{\mathbf{y} \in P_1} \|\mathbf{x} - \mathbf{y}\| \tag{6}$$

where $P_1$ and $P_2$ refer to any point clouds. By applying the self-supervised loss, we ensure that our network generates complete point clouds containing the shapes of the input point clouds in the real domain.

According to previous studies [3, 8, 12, 19, 20, 36, 38, 39, 48], the discriminator module has commonly been employed as a typical solution to mitigate domain gaps. In line with this, we additionally incorporate the discriminator module into our framework to provide supplementary support in bridging domain gaps. We employ discriminators $D$ for each part and train them within our entire framework. Discriminator loss, $\mathcal{L}_{disc}$, is define as:

$$\mathcal{L}_{disc} = -\lambda_4 \sum_{k=1}^{n} \log D(z_k^s) - \lambda_5 \sum_{k=1}^{n} [\log D(z_k^r) + \log(1 - D(z_k^r))] \tag{7}$$

where $\lambda_4$ and $\lambda_5$ are weight parameters for each term. By applying $\mathcal{L}_{disc}$, part-aware features can be induced toward a domain-invariant direction, thus providing additional support in reducing domain gaps.

In summary, we train our framework by using $\mathcal{L}_{comp}$, $\mathcal{L}_{self}$ and $\mathcal{L}_{disc}$. The total loss function $\mathcal{L}_{total}$ is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{comp} + \mathcal{L}_{self} + \mathcal{L}_{disc}. \tag{8}$$

## 4 Experiments

### 4.1 Datasets

We conduct experiments using ShapeNet [2] as the synthetic dataset and the USSPA dataset [19] as the real-world dataset. Part labels for synthetic point clouds are acquired from PartNet [21], which is also generated from ShapeNet [2]. Incomplete point clouds for synthetic domains are generated by randomly deleting certain regions from complete point clouds. Specifically, one point is randomly selected from the complete point cloud, and a certain amount of adjacent points are dropped to get an incomplete point cloud. The USSPA dataset [19] comprises scans from real scenes and their corresponding complete point clouds. The complete point clouds in the USSPA dataset [19] are selected from the synthetic dataset and closely resemble their corresponding real point clouds. Our framework is primarily evaluated on the USSPA dataset [19] as it provides GT for comparison.

**Table 1:** Quantitative results on USSPA dataset [19]. [CD↓/F1-score↑] are taken as the metric to evaluate the performance where CD is scaled with $\times 10^2$. Best results are indicated as **bold**.

| Setting | Method | Chair | Table | Bed | Lamp | Average |
|---|---|---|---|---|---|---|
| Supervised | PoinTr [46] | 7.73/75.9 | 7.99/78.9 | 9.21/67.3 | 6.90/77.4 | 7.96/74.9 |
| | Snowflake [42] | 8.56/73.5 | 8.70/77.0 | 8.06/76.3 | 7.05/75.5 | 8.09/75.6 |
| Syn-to-real | ShapeInv. [48] | 9.36/66.2 | 13.67/53.4 | 9.29/68.0 | 11.33/57.8 | 10.91/61.4 |
| | OptDE [8] | 13.42/52.9 | 21.28/23.0 | 11.60/55.9 | 17.14/38.6 | 15.86/42.6 |
| | USSPA [19] | 8.00/74.7 | 7.46/81.8 | 8.78/72.4 | 11.80/52.5 | 9.01/70.3 |
| | Ours | **7.63**/**76.6** | **6.54**/**83.8** | **6.98**/**82.2** | **6.80**/**79.2** | **6.99**/**80.5** |

**Table 2:** Quantitative results on USSPA dataset [19]. [EMD↓/MMD↓] are taken as teh metric to evaluate the performance where EMD is scaled with $\times 10$, and MMD is scaled with $\times 10^2$. Best results are indicated as **bold**.

| Method | Chair | Table | Bed | Lamp | Average |
|---|---|---|---|---|---|
| ShapeInv. [48] | 14.69/15.39 | 19.74/13.89 | **14.77**/18.54 | 19.79/12.17 | 17.25/15.00 |
| OptDE [8] | 26.64/14.40 | 29.87/**10.48** | 21.86/21.56 | 34.65/12.15 | 28.26/14.65 |
| USSPA [19] | 13.08/15.67 | 12.92/18.54 | 14.81/20.88 | 22.58/11.90 | 15.85/16.75 |
| Ours | **12.27**/**12.69** | **12.56**/15.18 | 16.77/**17.27** | **15.87**/**9.54** | **14.37**/**13.67** |

### 4.2 Implementation Details

Our framework is implemented on the PyTorch. The number N of input point clouds is 2048. We employ the pre-trained part segmentation network introduced in [34]. We conduct experiments on shared categories between the PartNet [21] and USSPA datasets [19] which are chair, table, bed, and lamp. Parts are merged from each category into {armrest, legs, seats, back}, {top, base}, {sleep area, frame, ladder}, and {body, base, unit}, respectively. To calculate the confidence score, we set the threshold in a range of 5% to 10% of the total points, ensuring a controlled dropping. More details are introduced in *Supp.* The weights $\lambda_{1\sim5}$ of training losses are set to $10^4$, $10^4$, $10^3$, 1 and 1, respectively. The batch size is 32, and the maximum epoch is 200 for chair and table and 500 for lamp and bed. We use AdamW optimizer, whose learning rate is $1.0 \times 10^{-3}$ for training.

### 4.3 Comparison with State-of-The-Art.

We compare our method against previous works to validate the efficacy of our network. Supervised setting refers to methods that are trained on only synthetic datasets and evaluated on real datasets. Syn-to-real setting methods utilize both synthetic and real datasets during training. To ensure a fair comparison, we train and evaluate all methods using official codes.

**Quantitative Results.** We evaluate the performance of various completion methods in terms of Chamfer Distance (CD) and F1-score, following previous works [19, 46] as shown in Tab. 1. Our method exhibits significantly enhanced

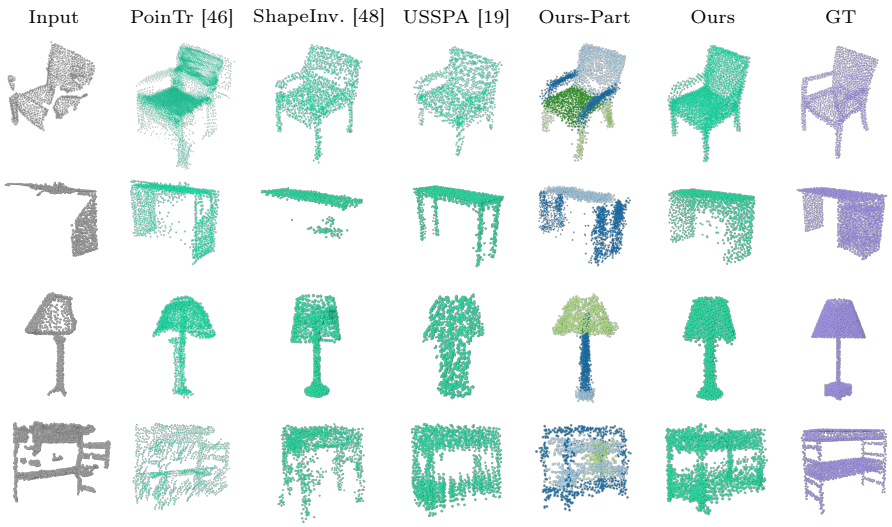


**Fig. 3:** Qualitative results on USSPA [19] dataset. From left to right: Input, PoinTr [46], ShapeInv. [48], USSPA [19], Ours-Part, Ours, and GT.

performance across all categories compared to previous methods. Specifically, our method achieves an average CD that is 2.02 lower and an average F1-score that is 10.17 higher than USSPA [19], which is state-of-the-art (SOTA) under the same setting. Additionally, we evaluate using Earth Mover Distance (EMD) and Minimal Matching Distance (MMD), following previous works [46, 48] as demonstrated in Tab. 2. Similarly, our method surpasses most categories and achieves a higher average score for each metric. These results demonstrate the superiority of our method over previous state-of-the-art approaches.

**Qualitative Results** We conduct a qualitative comparison of point cloud completion results using the USSPA dataset [19]. Figure 3 illustrates the results obtained from existing methods and our proposed method. The 'Ours-Part' results represent integrated part point clouds from the decoder, while the 'Ours' results depict the final complete point clouds from the refinement module. According to the 'Ours-Part' results, our method effectively completes each part. Subsequently, noisy regions are successfully improved, as shown in the 'Ours' results. Compared to other methods, our method demonstrates superior performance in point cloud completion, reflecting shape details from the input. For instance, in the second row of Fig. 3, other methods fail to complete missing regions or wrongly complete them. Conversely, our method successfully restores missing parts reflecting the input shape details. In summary, the qualitative results support that our method achieves substantial improvements in point cloud completion for real-world data.

**Table 3:** Effectiveness of the proposed modules. The numbers shown are [CD ↓/ F1-score ↑], where CD is scaled with $\times 10^2$. Best results are indicated as **bold**.

| Baseline | Disc. | Refine. | CD | F1-score |
|---|---|---|---|---|
| ✓ | | | 9.43 | 67.4 |
| ✓ | ✓ | | 8.96 | 69.2 |
| ✓ | | ✓ | 7.38 | 78.3 |
| ✓ | ✓ | ✓ | **6.99** | **80.4** |

**Table 4:** Effectiveness of the filtering process. The numbers shown are [CD ↓/ F1-score ↑], where CD is scaled with $\times 10^2$. Best results are indicated as **bold**.

| Filter | CD | F1-score |
|---|---|---|
| | 8.65 | 72.0 |
| ✓ | **7.51** | **77.0** |

**Table 5:** Quantitative results on USSPA dataset [19] for categories with cluster-based part labels. [CD↓/F1-score↑] are taken as the metric to evaluate the performance where CD is scaled with $\times 10^2$. Best results are indicated as **bold**.

| Setting | Method | Trash bin | TV | Cabinet | Bookshelf | Sofa | Tub | Average |
|---|---|---|---|---|---|---|---|---|
| Supervised | PoinTr [46] | 9.20/67.2 | 6.90/81.9 | 10.97/64.6 | 8.13/73.8 | 7.22/79.4 | 6.17/86.5 | 8.10/77.2 |
| | Snowflake [42] | 9.43/67.8 | 7.63/79.0 | 12.13/57.7 | 7.24/79.8 | 7.38/76.8 | 6.82/82.9 | 8.44/75.2 |
| Syn-to-real | ShapeInv. [48] | 9.92/64.4 | 8.21/77.5 | 11.75/60.7 | 7.90/75.8 | 7.39/81.6 | 6.82/83.3 | 8.66/75.8 |
| | OptDE [8] | 10.36/63.7 | 6.65/83.5 | 17.69/54.2 | 7.44/78.9 | 8.40/72.3 | 5.75/87.2 | 9.38/75.2 |
| | USSPA [19] | 10.67/56.9 | 7.66/77.7 | **10.74**/**67.4** | 7.86/77.6 | **6.50**/**88.0** | 6.74/82.0 | 8.36/78.6 |
| | Ours | **8.49**/**71.7** | **5.73**/**88.3** | 14.03/52.0 | **6.57**/**83.8** | 7.15/80.2 | **5.20**/**92.5** | **7.86**/**79.4** |

### 4.4 Ablation Studies

**Effectiveness of proposed modules** We conduct ablation studies to assess the effectiveness of each module introduced in our framework. Table 3 shows the performance changes computed by Chamfer Distance (CD) and F1-score from chair, table, bed, and lamp categories. The baseline framework comprises the Part-Based Decomposition (PBD) module, encoder *Enc*, part-aware transformer, and decoder *Dec*. Incorporating only the refinement module to the baseline results in a decrease of 2.05 in CD and an increase of 10.9 in F1-score. The addition of the discriminator module to the baseline framework leads to a decrease of 0.47 in CD and an increase of 1.8 in F1-score. As shown in the last row, the framework with all modules achieves the best performance in both CD and F1-score metrics. These results confirm the effectiveness of all proposed modules in point cloud completion for real-world scans. Furthermore, we can conclude that refinement module is more effective than discriminator.

**Effectiveness of filtering in real point clouds** To verify the effectiveness of the filtering process introduced in Sec. 3.2, we evaluate the performance with and without filtering, as shown in Tab. 4. Except for the filtering process in the real domain, we employ the same framework proposed in this paper. We evaluate the CD and F1-score for the chair, table, bed, and lamp categories and compare the average values. With the filtering process, CD decreases by 1.14 on average while F1-score increases by 5.0 on average compared to our framework without the filtering process. This result demonstrates the effectiveness of the filtering process for preventing negative effects from noisy points.

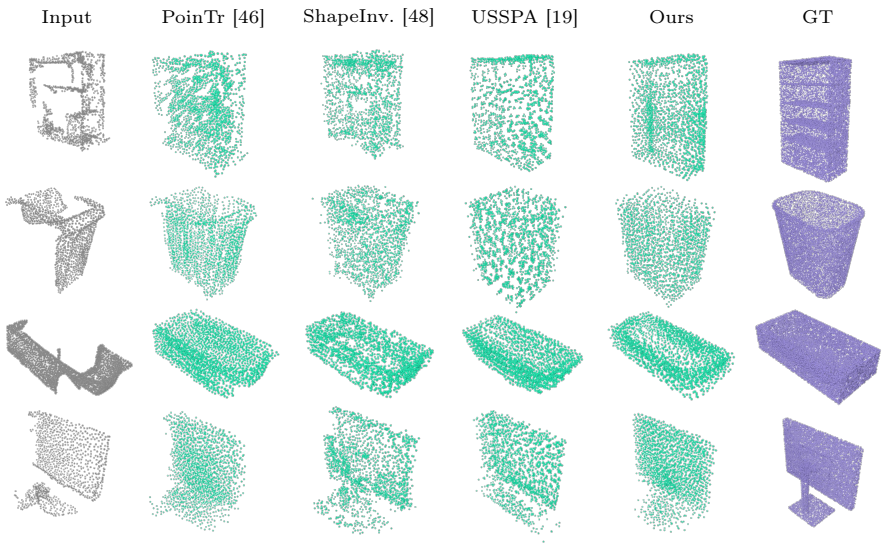


**Fig. 4:** Qualitative results on USSPA [19] dataset for categories without part labels. From left to right: Input, PoinTr [46], ShapeInv. [48], USSPA [19], Ours, and GT.

### 4.5 Additional Analysis

**Analysis of part-based framework** While verifying the effectiveness of modules as shown in Sec. 4.4, we have observed notable results from Tab. 3. Methods in first and third row of Tab. 3 do not contain any discriminator module. In other words, only the part-based framework alone serves as the key element in bridging the domain gap. Nevertheless, results from the baseline are comparable to USSPA [19], and results from the baseline with the refinement module outperform all methods. This suggests that our proposed part-based framework effectively bridges the domain gap without getting help from discriminator modules which are commonly used. Therefore, we can conclude that our main idea effectively facilitates domain adaptation in the task of point cloud completion.
**Categories without part labels** Some categories lack part labels even in the synthetic domain, which limits the practical applicability of our methods. To address this challenge, we employ a cluster-based co-segmentation approach, similar to previous works such as [24, 27], which can be utilized for categories lacking part labels. Through the cluster-based co-segmentation approach, we decompose the remaining categories in the USSPA [19] and ShapeNet [2] datasets. Further details on the co-segmentation process are provided in the *Supp.*

We train and evaluate our method on the trash bin, TV, cabinet, bookshelf, sofa, and tub categories. Table 5 shows the quantitative results, including comparisons with other methods using Chamfer Distance (CD) and F1-score metrics. Across the experimented categories, our method outperforms existing methods for both metrics. Additionally, our method still demonstrates state-of-

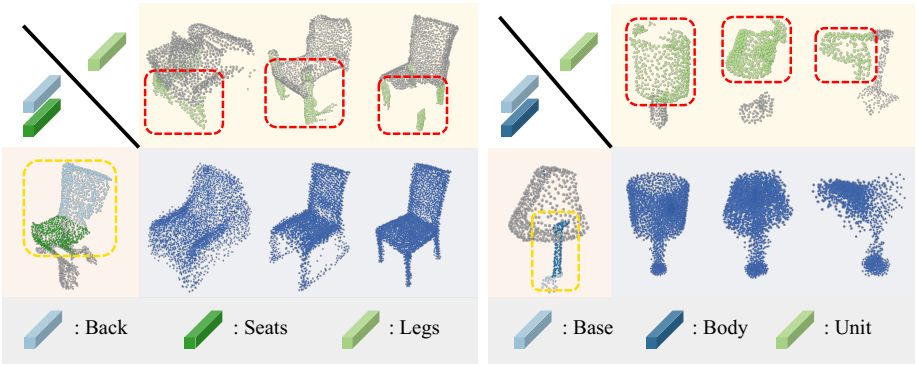


**Fig. 5:** Visualization of part manipulation experiments. Point clouds on the left side are reference, and the upper side are point clouds for feature swapping. Used parts are highlighted with dashed boxes. Predicted point clouds after feature manipulation are located in the sky-blue region. The color of each part is depicted on the bottom side.

the-art (SOTA) results in terms of average CD and F1-score. Figure 4 illustrates qualitative results for the bookshelf, trash bin, tub, and TV categories. Our method successfully produces complete point clouds corresponding to the input, effectively reconstructing missing regions with high quality compared to other methods. These results suggest the robustness and applicability of our method even in cases where part labels are not available.

**Part feature manipulation** As explained in Sec. 3.3, we design the part features, $\{f_k^r\}_{k=1}^n$, to encapsulate the semantic and geometric information of each shape, while the part-aware features, $\{z_k^r\}_{k=1}^n$, capture part relationships for inferring missing regions. To verify whether PAC is trained following our intention, we experiment with manipulating part features. We first select one incomplete point cloud as a reference and extract part features from incomplete point clouds, including the reference point cloud. Then, we replace certain part feature of reference point cloud with corresponding part feature from another point cloud and then generate complete point cloud. If our framework is trained as intended, each shape from the utilized part should be reflected while missing parts are restored.

Figure 5 illustrates the used incomplete point clouds and the predicted complete point clouds. The left side shows the reference point cloud, while the upper side shows the point clouds for swapping part features, indicated by dashed boxes. The completed point clouds, located in sky-blue region, show shapes reflecting utilized parts following our expectations. Based on this, we conclude that our framework has successfully learned the concept of part as our intention.

**Qualitative results on ScanObjectNN** We additionally conduct an experiment using the ScanObjectNN dataset [30], which is also derived from real-world scans. As this dataset lacks complete point clouds corresponding to each real-world point cloud, direct evaluation with ground truth is not possible. Instead, we conduct a qualitative analysis by visualizing our predictions in Fig. 6. The 'Ours-Part' results represent integrated part-wise predictions, while the 'Ours'

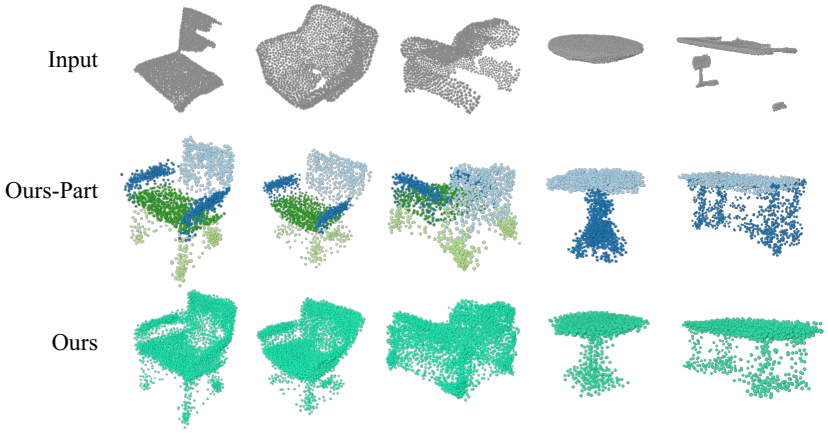


**Fig. 6:** Qualitative results on ScanObjectNN dataset [30] for chairs and tables. From up to bottom: Input, Ours-Part, and Ours.

results depict the final refined predictions. We observe that our method generates part-wise predictions with detailed shapes from incomplete inputs. Furthermore, the refined predictions show improvement by addressing noise present in the part-wise predictions. These results illustrate the applicability and effectiveness of our method across various real-world datasets.

## 5 Conclusion

We propose a novel framework that harnesses parts as a pivotal element for synthetic-to-real point cloud completion. Our method employs a Part-Based Decomposition (PBD) module, facilitating consistent point cloud decomposition across both synthetic and real domains. Subsequently, we introduce a Part-Aware Completion (PAC) module designed to address completion tasks in both domains. To facilitate the learning of part relationships crucial for completion, our PAC module incorporates a novel part-aware transformer. Furthermore, we incorporate a refinement module aimed at enhancing noisy shapes within the completed point cloud. Our experimental results demonstrate the state-of-the-art performance of the proposed method across various metrics when applied to real-world data. Moreover, we conduct experiments on categories lacking part labels to show the practicality of our approach. We believe that our framework showcases the effectiveness of the "part" which proves to be highly effective in the syn-to-real point cloud completion.

**Limitation** We have observed errors in the synthetic dataset, such as the objects appeared inconsistent with their respective categories. Furthermore, both synthetic and real-world datasets are limited in their capacity for point cloud completion. We believe that our method could yield improved performance given access to larger, error-free datasets.

**Acknowledgements** This work was supported by the Technology Innovation Program (1415187329, 20024355, Development of autonomous driving connectivity technology based on sensor-infrastructure cooperation) funded by the Ministry of Trade, Industry & Energy(MOTIE, Korea) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF2022R1A2B5B03002636).

## References

1. Bokhovkin, A., Dai, A.: Neural part priors: Learning to optimize part-based object completion in rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9032–9042 (2023)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
3. Chen, X., Chen, B., Mitra, N.J.: Unpaired point cloud completion on real scans using adversarial training. arXiv preprint arXiv:1904.00069 (2019)
4. Chen, Z., Long, F., Qiu, Z., Yao, T., Zhou, W., Luo, J., Mei, T.: Anchorformer: Point cloud completion from discriminative nodes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13581–13590 (2023)
5. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6970–6981 (2020)
6. Cui, R., Qiu, S., Anwar, S., Liu, J., Xing, C., Zhang, J., Barnes, N.: P2c: Self-supervised point cloud completion from single partial clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14351–14360 (2023)
7. Fu, Z., Wang, L., Xu, L., Wang, Z., Laga, H., Guo, Y., Boussaid, F., Bennamoun, M.: Vapcnet: Viewpoint-aware 3d point cloud completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12108–12118 (2023)
8. Gong, J., Liu, F., Xu, J., Wang, M., Tan, X., Zhang, Z., Yi, R., Song, H., Xie, Y., Ma, L.: Optimization over disentangled encoding: Unsupervised cross-domain point cloud completion via occlusion factor manipulation. In: European Conference on Computer Vision. pp. 517–533. Springer (2022)
9. Gu, J., Ma, W.C., Manivasagam, S., Zeng, W., Wang, Z., Xiong, Y., Su, H., Urtasun, R.: 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. pp. 283–299. Springer (2020)
10. Hong, S., Yavartanoo, M., Neshatavar, R., Lee, K.M.: Acl-spc: Adaptive closed-loop system for self-supervised point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9435–9444 (2023)
11. Hu, T., Han, Z., Zwicker, M.: 3d shape completion with multi-view consistent inference. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10997–11004 (2020)
12. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. Advances in neural information processing systems **31** (2018)

13. Kim, J., Kweon, H., Yang, Y., Yoon, K.J.: Learning point cloud completion without complete point clouds: A pose-aware approach. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14203–14213 (October 2023)
14. Li, G., Chen, Y., Cheng, M., Wang, C., Li, J.: N-dpc: Dense 3d point cloud completion based on improved multi-stage network. In: Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition. pp. 274–279 (2020)
15. Li, S., Gao, P., Tan, X., Wei, M.: Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9466–9475 (2023)
16. Liu, J., Mahdavi-Amiri, A., Savva, M.: Paris: Part-level reconstruction and motion analysis for articulated objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 352–363 (2023)
17. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M.: Morphing and sampling network for dense point cloud completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11596–11603 (2020)
18. Liu, W., Wu, Y., Ruan, S., Chirikjian, G.S.: Marching-primitives: Shape abstraction from signed distance function. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8771–8780 (2023)
19. Ma, C., Chen, Y., Guo, P., Guo, J., Wang, C., Guo, Y.: Symmetric shape-preserving autoencoder for unsupervised real scene point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13560–13569 (2023)
20. Ma, C., Yang, Y., Guo, J., Pan, F., Wang, C., Guo, Y.: Unsupervised point cloud completion and segmentation by generative adversarial autoencoding network. Advances in Neural Information Processing Systems **35**, 3556–3568 (2022)
21. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: 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. pp. 909–918 (2019)
22. Nakayama, G.K., Uy, M.A., Huang, J., Hu, S.M., Li, K., Guibas, L.: Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14257–14267 (2023)
23. Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S., Zhang, J., et al.: Skeleton-bridged point completion: From global inference to local adjustment. Advances in Neural Information Processing Systems **33**, 16119–16130 (2020)
24. Sajnani, R., Poulenard, A., Jain, J., Dua, R., Guibas, L.J., Sridhar, S.: Condor: Self-supervised canonicalization of 3d pose for partial shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16969–16979 (2022)
25. Shen, Y., Feng, C., Yang, Y., Tian, D.: Mining point cloud local structures by kernel correlation and graph pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4548–4557 (2018)
26. Shuai, Q., Zhang, C., Yang, K., Chen, X.: Dpf-net: Combining explicit shape priors in deformable primitive field for unsupervised structural reconstruction of 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14321–14329 (2023)

27. Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G., Yi, K.M.: Canonical capsules: Unsupervised capsules in canonical pose. arXiv preprint arXiv:2012.04718 **1** (2021)
28. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 383–392 (2019)
29. Tertikas, K., Paschalidou, D., Pan, B., Park, J.J., Uy, M.A., Emiris, I., Avrithis, Y., Guibas, L.: Generating part-aware editable 3d shapes without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4466–4478 (2023)
30. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1588–1597 (2019)
31. Wang, W., Zhang, R., You, M., Zhou, H., He, B.: 3d assembly completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2663–2671 (2023)
32. Wang, X., Ang, M.H., Lee, G.H.: Voxel-based network for shape completion by leveraging edge generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13189–13198 (2021)
33. Wang, Y., Tan, D.J., Navab, N., Tombari, F.: Learning local displacements for point cloud completion. In: Proceedings of thez IEEE/CVF conference on computer vision and pattern recognition. pp. 1568–1577 (2022)
34. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)
35. Wei, X., Gu, X., Sun, J.: Learning generalizable part-based feature representation for 3d point clouds. Advances in Neural Information Processing Systems **35**, 29305–29318 (2022)
36. Wen, X., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: 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. pp. 13080–13089 (2021)
37. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1939–1948 (2020)
38. Wu, R., Chen, X., Zhuang, Y., Chen, B.: Multimodal shape completion via conditional generative adversarial networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. pp. 281–296. Springer (2020)
39. Wu, Y., Yan, Z., Chen, C., Wei, L., Li, X., Li, G., Li, Y., Cui, S., Han, X.: Scoda: Domain adaptive shape completion for real scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17630–17641 (2023)
40. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
41. Xia, Y., Xia, Y., Li, W., Song, R., Cao, K., Stilla, U.: Asfm-net: Asymmetrical siamese feature matching network for point completion. In: Proceedings of the 29th ACM international conference on multimedia. pp. 1938–1947 (2021)

42. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5499–5509 (2021)
43. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: Gridding residual network for dense point cloud completion. In: European Conference on Computer Vision. pp. 365–381. Springer (2020)
44. Xu, X., Guerrero, P., Fisher, M., Chaudhuri, S., Ritchie, D.: Unsupervised 3d shape reconstruction by part retrieval and assembly. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8559–8567 (2023)
45. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 206–215 (2018)
46. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12498–12507 (2021)
47. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 international conference on 3D vision (3DV). pp. 728–737. IEEE (2018)
48. Zhang, J., Chen, X., Cai, Z., Pan, L., Zhao, H., Yi, S., Yeo, C.K., Dai, B., Loy, C.C.: Unsupervised 3d shape completion through gan inversion. In: CVPR (2021)
49. Zhang, X., Feng, Y., Li, S., Zou, C., Wan, H., Zhao, X., Guo, Y., Gao, Y.: View-guided point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15890–15899 (2021)
50. Zhu, Z., Chen, H., He, X., Wang, W., Qin, J., Wei, M.: Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14508–14518 (2023)
51. Zong, D., Sun, S., Zhao, J.: Ashf-net: Adaptive sampling and hierarchical folding network for robust point cloud completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3625–3632 (2021)