

Cycle4Completion: Unpaired Point Cloud Completion using Cycle Transformation with Missing Region Coding

Xin Wen¹, Zhizhong Han², Yan-Pei Cao³, Pengfei Wan³, Wen Zheng³, Yu-Shen Liu^{1*}

¹School of Software, BNRist, Tsinghua University, Beijing, China

²Department of Computer Science, Wayne State University, USA

³Y-tech, Kuaishou Technology, Beijing, China

x-wen16@mails.tsinghua.edu.cn h312h@wayne.edu caoyanpei@gmail.com

{wanpengfei,zhengwen}@kuaishou.com liuyushen@tsinghua.edu.cn

Abstract

In this paper, we present a novel unpaired point cloud completion network, named Cycle4Completion, to infer the complete geometries from a partial 3D object. Previous unpaired completion methods merely focus on the learning of geometric correspondence from incomplete shapes to complete shapes, and ignore the learning in the reverse direction, which makes them suffer from low completion accuracy due to the limited 3D shape understanding ability. To address this problem, we propose two simultaneous cycle transformations between the latent spaces of complete shapes and incomplete ones. Specifically, the first cycle transforms shapes from incomplete domain to complete domain, and then projects them back to the incomplete domain. This process learns the geometric characteristic of complete shapes, and maintains the shape consistency between the complete prediction and the incomplete input. Similarly, the inverse cycle transformation starts from complete domain to incomplete domain, and goes back to complete domain to learn the characteristic of incomplete shapes. We experimentally show that our model with the learned bidirectional geometry correspondence outperforms state-of-the-art unpaired completion methods. Code will be available at <https://github.com/diviswen/Cycle4Completion>.

1. Introduction

Point clouds, as a popular 3D representation, can be easily produced by 3D scanning devices and depth cameras. However, due to the limitations of the view angles of cam-

era/scanning devices and self-occlusion, raw point clouds are often sparse, noisy and partial, which usually require shape completion before being analyzed in further applications such as shape classification [27, 14, 23, 21, 24], retrieval [11, 7, 12], semantic/instance segmentation [22, 39]. Although the recent data-driven supervised completion methods [16, 46, 42, 44, 17, 20] have achieved impressive performance, they heavily rely on the paired training data, which consists of incomplete shapes and their corresponding complete ground truth. In real-world applications, however, such high quality and large-scale paired training dataset is not easy to access, which makes it hard to directly train a supervised completion network.

A promising but challenging solution to this problem is to learn a completion network in an unpaired way, where the common practice is to establish the shape correspondence between the incomplete shapes and complete ones from the unpaired training data without requiring the incomplete and complete correspondence. The latest work like Pcl2Pcl [2] introduced an adversarial framework to merge the geometric gap between the complete shape distribution and incomplete one in the latent representation space. Although many efforts have been made to learn the geometric correspondence from incomplete shapes to complete ones, previous methods ignore the inverse correspondence from complete shapes to incomplete ones, which leads to low completion accuracy due to the limited 3D shape understanding ability.

To address this problem, we propose a novel unpaired point cloud completion network, named Cycle4Completion, to establish the geometric correspondence between incomplete and complete shapes in both directions. We achieve this by designing two *cycle transformations*, i.e. the incomplete cycle transformation (incomplete-cycle) and the complete cycle transformation (complete-cycle), as shown in Figure 1. The incomplete-cycle in Figure 1(a) learns the mapping from the incomplete domain to the com-

*Corresponding author. This work was supported by National Key R&D Program of China (2020YFF0304100, 2018YFB0505400), the National Natural Science Foundation of China (62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data.

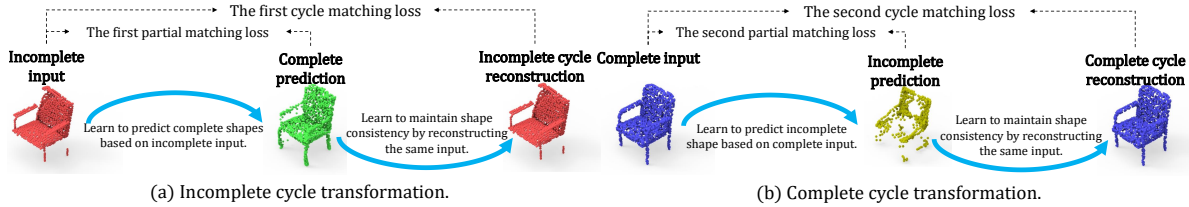


Figure 1. Illustration of cycle transformation, which consists of two inverse cycles, as shown in (a) and (b). The cycle transformation promotes network to understand 3D shapes by learning to generate complete or incomplete shapes from their complementary ones.

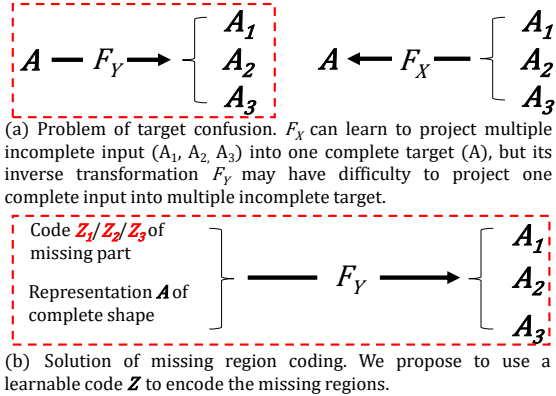


Figure 2. Illustration of target confusion and the solution of missing region coding.

plete one, which is then projected back to the incomplete domain. On the other hand, the complete-cycle in Figure 1(b) provides the completion knowledge on the inverse direction with incomplete input, which can be used to further enhance the incompleteness quality for incomplete-cycle. However, as shown in Figure 2(a), directly applying a cycle transformation in the latent space will encounter a new problem which we name it as the *target confusion* problem. This problem is raised when establishing shape correspondence from multiple incomplete shapes (e.g. A_1, A_2 and A_3) to one complete shape (e.g. A). This is because one of the cycle requires the network to predict the incomplete shape based on the complete input, and the corresponding transformation network F_Y cannot fully map one complete input into multiple different incomplete targets only through a deep neural network. To solve this problem, we propose the learnable *missing region coding* (MRC) to transform incomplete shapes to complete ones, as shown in Figure 2(b). The representations of incomplete shapes can be decomposed into two parts: one is the representation A of their corresponding complete shape, and the other one is the code Z to encode their missing regions. When predicting the complete shapes from the incomplete ones, only the representation A is considered, and when predicting the incomplete shapes from the complete ones, both the representation A and code Z are considered. Thus, the transformation network F_Y will relieve the confusion by learning to project one complete input to several incomplete targets. Instead, the learnable missing region code Z can help the

network clarify which incomplete shape is the current target for transformation, and relieve the target confusion problem. Our main contributions are summarized as follows.

- We propose a novel unpaired point cloud completion network, named Cycle4Completion. Compared with previous unpaired completion methods which only consider the single-side correspondence from incomplete shapes to complete ones, Cycle4Completion can enhance the completion performance by establishing the geometric correspondence between complete shapes and incomplete shapes from both directions.
- We propose the *cycle transformation* framework in latent space, which is combined with the *partial matching loss* and *cycle matching loss* to establish the bidirectional geometric correspondence between the complete and incomplete shapes, and maintain the shape consistency throughout the whole transformation process.
- We propose the *missing region coding* to decompose the incomplete shape representation into a representation of its corresponding complete shape, and a missing region code to encode the missing regions of the incomplete shapes, respectively. This solves the *target confusion* when the network tries to predict multiple incomplete shapes based on a single complete shape.

2. Related Work

3D shape completion has drawn an increasing attention in recent years [45, 40, 35]. Previous completion methods can be roughly divided into two categories, i.e. traditional approaches and deep learning based approaches, which we will detail below.

Traditional approaches for 3D shape completion. The traditional geometry/statistic based methods [32, 1, 36, 29, 26, 30] exploit the geometric features of surface on the partial input to generate the missing regions of 3D shapes [32, 1, 34, 36], or exploit the large-scale shape database to search for the similar shapes/patches to fill the missing regions of 3D shapes [29, 19, 26, 30]. For example, Hu et al.[36] proposed to exploit both the local smoothness and the non-local self-similarity in point clouds, by defining the smoothing and denoising properties of point clouds

and globally searching the similar area for the missing region. On the other hand, the data-driven shape completion methods like Shen et al.[30] formulate the completion of 3D shapes as a bottom up part assembling process, where a 3D shape repository is adopted as the reference to recover a variety of high-level complete structures. In all, these traditional shape completion approaches are mainly based on the hand-crafted rules to describe the characteristics of missing region, and the similarities between the missing region and complete shape. Therefore, the generalization ability of such kind of methods is usually limited. For example, the method proposed by Sung et al.[32] predefines several categories of semantic parts of 3D shapes, and uses geometric characteristics such as part positions, scales, and orientations to find similar parts for missing regions from shape database. Such kind of methods usually fails in the situation of more complicated shapes, which are beyond the description of the predefined semantic part categories or geometric characteristics. In contrast, deep learning based completion methods can learn more flexible features to predict a complete shape from an incomplete input. This kind of methods will be detailed in the subsection below.

Deep learning approaches for 3D shape completion. The second category includes neural networks based methods, which take advantage of deep learning to learn the representation from the input shape [12, 38, 8, 22, 18] and predict the complete shape according to the representation, using an encoder-decoder framework. This category can be further classified according to different input shape forms including: volumetric shape completion [3, 5, 31] and point cloud completion [13, 33, 28, 15, 35, 37]. Our Cycle4Completion also falls into this category, which completes 3D shapes represented by point clouds. Notable recent studies like CDN [35], NSFA [47] and SA-Net [40] have achieved impressive results on supervised point cloud completion task. Moreover, RL-GAN-Net [28] introduced the reinforcement learning with the adversarial training to further improve the reality and consistency of the generated complete point clouds. However, although great improvements have been made in supervised point cloud completion task, this task strongly depends on the paired training data, but the paired ground truth for incomplete real-world scan is rarely available. On the other hand, there is very few studies concerning the unpaired point cloud completion task. As one of the pioneering work, AML [31] directly measured the maximum likelihood between the latent representation of incomplete and complete shapes. Following the similar practice, Pcl2Pcl [2] introduced the GAN framework to bridge the semantic gap between incomplete and complete shapes. And Wu et al. [41] proposed a VAE based framework to predict multiple complete shapes for a single incomplete input.

Compared with the above-mentioned unpaired methods, our Cycle4Completion further establishes the self-

supervision by cycle transformations in the latent space from both directions, which can provide a better guidance to learn the bidirectional geometric correspondence between incomplete shape and complete ones.

Relationships with GANs. Our work is also related to the generative adversarial networks (GAN). Especially, our work is inspired by the unpaired style transferring network CycleGAN [48] in 2D domain. However, it is usually difficult to directly apply a framework like CycleGAN to point cloud completion, where the simple cycle-consistency loss often fails to guide the generator to infer the missing shapes, because conceiving a consistent missing shape for the incomplete input is more complicated than transferring styles. Therefore, we propose to perform the cycle transformation in the latent space, where the partial and cycle matching losses are proposed for maintaining the transferred shapes consistency. Considering that 3D completion is essentially a reconstruction process from 3D shape to 3D shapes, the reconstruction of 3D shapes from 2D images [9, 6, 10] is also a notable research direction, which is closely related to 3D completion. The difference between the two tasks is that 3D reconstruction from 2D images does not require 3D information as input, while the completion task based on 3D shapes requires 3D shape information as input.

3. The Architecture of Cycle4Completion

3.1. Formulation

We first describe the basic formulations in our method. As shown in Figure 3(a), let $\mathcal{P}_X = \{\mathbf{p}_i^x\}$ denote the point cloud of an incomplete shape, and $\mathcal{P}_Y = \{\mathbf{p}_i^y\}$ denote the point cloud of complete one. Our goal is to learn two mappings F_X and F_Y between the latent representations $\{\mathbf{x}\}$ of incomplete shapes and the latent representations $\{\mathbf{y}\}$ of complete shapes. These representations are generated by the point cloud encoders $E_X : \mathcal{P}_X \rightarrow \mathbf{x}$ and $E_Y : \mathcal{P}_Y \rightarrow \mathbf{y}$, respectively, which are trained under the auto-encoder framework with the point cloud generators G_X and G_Y , respectively. In addition, two adversarial discriminators D_X and D_Y are introduced. D_X aims to distinguish between $\{\mathbf{x}\}$ and $\{\mathbf{y}_x\}$, where $\mathbf{y}_x = F_Y(\mathbf{y})$. D_Y aims to distinguish between $\{\mathbf{y}\}$ and $\{\mathbf{x}_y\}$, where $\mathbf{x}_y = F_X(\mathbf{x})$. We denote the compound operation of two functions F_X and F_Y as $F_X F_Y$.

3.2. Encoder-decoder for Learning Latent Space

The two auto-encoders learn the latent representation spaces for incomplete and complete shapes, respectively. We define the full Chamfer distance (CD) between two point clouds \mathcal{P}_1 and \mathcal{P}_2 as

$$L_{CD}(\mathcal{P}_1 \rightleftharpoons \mathcal{P}_2) = \sum_{\mathbf{p}_i^1 \in \mathcal{P}_1} \min_{\mathbf{p}_i^2 \in \mathcal{P}_2} \|\mathbf{p}_i^1 - \mathbf{p}_i^2\| + \sum_{\mathbf{p}_i^2 \in \mathcal{P}_2} \min_{\mathbf{p}_i^1 \in \mathcal{P}_1} \|\mathbf{p}_i^2 - \mathbf{p}_i^1\|. \quad (1)$$

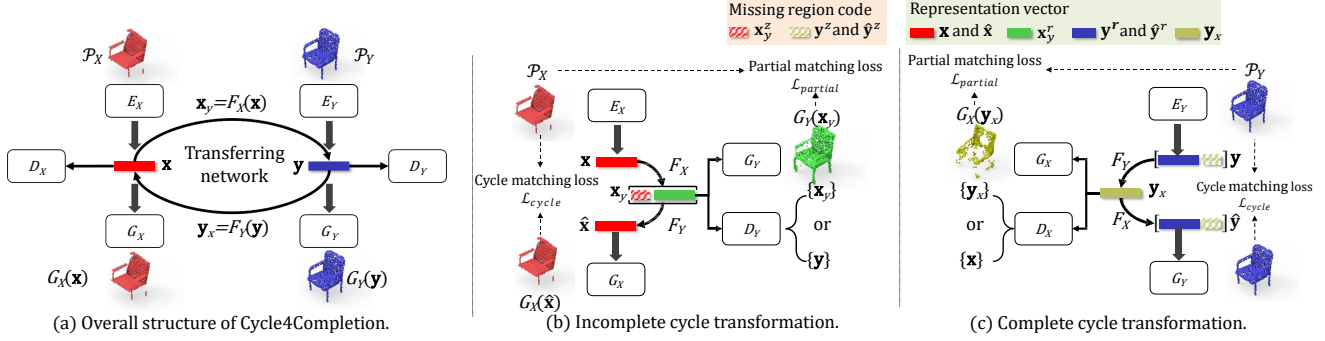


Figure 3. Illustration of Cycle4Completion. The overall structure in (a) consists of the incomplete cycle transformation in (b), which produces the complete prediction (green) from the incomplete input (red), and the complete cycle transformation in (c), which produces the incomplete prediction (yellow) from the complete input (blue). Both of the two cycles use self-reconstruction to learn the shape consistency.

The reconstruction loss \mathcal{L}_{AE} for training the auto-encoder framework is formulated as:

$$\mathcal{L}_{AE} = \mathcal{L}_{CD}(\mathcal{P}_X \rightleftharpoons G_X(\mathbf{x})) + \mathcal{L}_{CD}(\mathcal{P}_Y \rightleftharpoons G_Y(\mathbf{y})). \quad (2)$$

3.3. Cycle Transformation

Transformation with missing region coding. For the incomplete cycle transformation in Figure 3(b), the missing region code \mathbf{x}_y^z and the complete shape representation \mathbf{x}_y^r are generated by F_X when transferring \mathbf{x} from the incomplete domain into the complete domain as \mathbf{x}_y . Therefore, \mathbf{x}_y can be further denoted as $\mathbf{x}_y = [\mathbf{x}_y^r : \mathbf{x}_y^z]$. The remark “:” indicates the concatenation of two feature vectors. The complete shape is then predicted based on \mathbf{x}_y^r by G_Y as $G_Y(\mathbf{x}_y^r)$. And the discriminator D_Y only discriminates between \mathbf{x}_y^r and \mathbf{y} . In order to establish the shape consistency during the transformation process, \mathbf{x}_y is projected back into the incomplete domain again by F_Y , denoted as $\hat{\mathbf{x}}$. The cycle reconstructed shape is predicted by G_X , denoted as $G_X(\hat{\mathbf{x}})$.

For the complete cycle transformation in Figure 3(c), the encoder E_Y directly predicts a complete shape representation \mathbf{y}^r . In order to predict an incomplete shape, we randomly sample a missing region code from a uniform distribution between $[0, 1]$, denoted as \mathbf{y}^z , and concatenate it with \mathbf{y}^r , denoted as $\mathbf{y} = [\mathbf{y}^r : \mathbf{y}^z]$. Then, the transformation network F_Y transforms \mathbf{y} into the incomplete domain, denoted as \mathbf{y}_x . Similar to the incomplete cycle transformation, the incomplete shape is predicted based on \mathbf{y}_x by G_X , denoted as $G_X(\mathbf{y}_x)$. And the discriminator D_X discriminates between \mathbf{y}_x and \mathbf{x} . Following the inverse direction of incomplete cycle transformation, the shape consistency during the complete cycle transformation is established by predicting the reconstructed shape $G_Y(\hat{\mathbf{y}})$, where $\hat{\mathbf{y}} = F_X(\mathbf{y}_x)$. Note that same as \mathbf{y} , $\hat{\mathbf{y}}$ also consists of a complete representation $\hat{\mathbf{y}}^r$ and a missing region code $\hat{\mathbf{y}}^z$.

Code matching Loss. In the complete cycle transformation in Figure 3(c), a missing region code \mathbf{y}^z is sampled from a uniform distribution in order to create missing regions from

the current complete input \mathcal{P}_Y . After the shape \mathcal{P}_Y is cycled through F_Y and F_X , a new missing region code $\hat{\mathbf{y}}^z$ is predicted by the transformation network $F_Y F_X$. Because both \mathbf{y}^z and $\hat{\mathbf{y}}^z$ correspond to the same incomplete shape, the two codes should be equal. Therefore, we propose to use the Euclidean distance between \mathbf{y}^z and $\hat{\mathbf{y}}^z$ as the *code matching loss*, which can be formulated as:

$$\mathcal{L}_{code} = \|\mathbf{y}^z - \hat{\mathbf{y}}^z\|^2. \quad (3)$$

Cycle matching loss. The *cycle matching loss* aims to match the shapes of cycle reconstruction $G_Y(\hat{\mathbf{y}})/G_X(\hat{\mathbf{x}})$ to their corresponding input $\mathcal{P}_Y/\mathcal{P}_X$, which should keep the shape consistency throughout the whole transformation process. Specifically, we define the cycle matching loss as the full Chamfer distance between the input $\mathcal{P}_Y/\mathcal{P}_X$ and the reconstructed point cloud $G_Y(\hat{\mathbf{y}})/G_X(\hat{\mathbf{x}})$ as $\mathcal{L}_{CD}(\mathcal{P}_X \rightleftharpoons G_X(\hat{\mathbf{x}}))$ and $\mathcal{L}_{CD}(\mathcal{P}_Y \rightleftharpoons G_Y(\hat{\mathbf{y}}))$, respectively. Then we indicate the *full cycle matching loss* for transferring network F_X and F_Y as:

$$\mathcal{L}_{cycle} = \mathcal{L}_{CD}(\mathcal{P}_X \rightleftharpoons G_X(\hat{\mathbf{x}})) + \mathcal{L}_{CD}(\mathcal{P}_Y \rightleftharpoons G_Y(\hat{\mathbf{y}})). \quad (4)$$

Partial matching loss. The *partial matching loss* is a directional constraint, which aims to match one shape to another without the matching in the inverse direction. Similar practice can be found in previous work [2], which adopted the directional Hausdorff distance to partially match the complete prediction to the incomplete input. However, the partial matching on the single direction cannot provide further guidance for the inference of missing regions, so we integrate the partial matching into the cycle transformation to establish a more comprehensive geometric correspondence on both directions. We define the partial Chamfer distance between two point clouds \mathcal{P}_1 and \mathcal{P}_2 as:

$$\mathcal{L}_{CD'}(\mathcal{P}_1 \rightarrow \mathcal{P}_2) = \sum_{\mathbf{p}_i^1 \in \mathcal{P}_1} \min_{\mathbf{p}_i^2 \in \mathcal{P}_2} \|\mathbf{p}_i^1 - \mathbf{p}_i^2\|. \quad (5)$$

It is a constraint that only requires that the shape of \mathcal{P}_2 partially matches the shape of \mathcal{P}_1 . For incomplete-

cycle in Figure 3(b), the partial matching loss is formulated as $\mathcal{L}_{CD'}(\mathcal{P}_X \rightarrow G_Y(\mathbf{x}_y^r))$, and for complete-cycle in Figure 3(c), the partial matching loss is formulated as $\mathcal{L}_{CD'}(G_X(\mathbf{y}_x) \rightarrow \mathcal{P}_Y)$. Note that the directions of above two partial Chamfer distances are always pointed from incomplete shapes to complete ones, which guarantees the incomplete shape partially matches the complete one, no matter whether it is predicted or real. The *full partial matching loss* is defined as:

$$\mathcal{L}_{\text{partial}} = \mathcal{L}_{CD'}(\mathcal{P}_X \rightarrow G_Y(\mathbf{x}_y^r)) + \mathcal{L}_{CD'}(G_X(\mathbf{y}_x) \rightarrow \mathcal{P}_Y). \quad (6)$$

Adversarial loss. To further bridge the geometric gap between the latent representations of complete and incomplete shapes, the adversarial learning framework is adopted as an unpaired constraint. Specifically, two discriminators D_X and D_Y are used to distinguish the real and fake representations in the incomplete and complete domains, respectively. The D_X in incomplete domain discriminates between the real latent representations $\{\mathbf{x}\}$ and the fake latent representations $\{\mathbf{y}_x\}$; in the same way, the D_Y in complete domain discriminates between $\{\mathbf{y}\}$ and $\{\mathbf{x}_y\}$. In order to stabilize the training, we formulate the objective loss for discriminator under the WGAN-GP [4] framework. For simplicity, we formulate the loss for D_X as:

$$\mathcal{L}_{D_X} = \mathbb{E}_{\mathbf{x}} D_X(\mathbf{x}) - \mathbb{E}_{\mathbf{y}_x} D_X(\mathbf{y}_x) + \lambda_{gp} \mathcal{T}_{D_X}, \quad (7)$$

where λ_{gp} is a pre-defined weight factor and \mathcal{T}_{D_X} is gradient penalty term, denoted as:

$$\mathcal{T}_{D_X} = \mathbb{E}_{\mathbf{x}} [(\|\nabla_{\mathbf{x}} D_X(\mathbf{x})\|_2 - 1)^2]. \quad (8)$$

The discriminator loss \mathcal{L}_{D_Y} for D_Y can be formulated in the same way. The final adversarial losses for generator $\{F_X, F_Y\}$ and discriminator $\{D_X, D_Y\}$ are given as

$$\mathcal{L}_D = \mathcal{L}_{D_X} + \mathcal{L}_{D_Y}, \quad (9)$$

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{y}_x} D_X(\mathbf{y}_x) + \mathbb{E}_{\mathbf{x}_y} D_Y(\mathbf{x}_y^r). \quad (10)$$

3.4. Training Strategy

In our model, there are four sets of losses in total. We use Θ_D to denote the trainable parameters in $\{D_X, D_Y\}$, Θ_{AE} to denote the trainable parameters in $\{E_X, G_X, E_Y, G_Y\}$, and Θ_F to denote the trainable parameters in $\{F_X, F_Y\}$. We use $\mathcal{L}_G(\Theta_{AE}, \Theta_F, \Theta_D)$ to denote that there are three parts of network (i.e. auto-encoder, transferring network, and discriminator) involved in calculating \mathcal{L}_G .

Given the learning rate γ , the encoder-decoder loss regularizes the parameters Θ_{AE} , where the gradient optimization step is expressed as

$$\Theta_{AE} \leftarrow \Theta_{AE} - \gamma \frac{\partial \mathcal{L}_{AE}(\Theta_{AE})}{\partial \Theta_{AE}}. \quad (11)$$

The cycle matching loss and partial matching loss along with the adversarial loss regularize the transferring network. The gradient descent step for Θ_F is given as

$$\Theta_F \leftarrow \Theta_F - \gamma [\lambda_g \frac{\partial \mathcal{L}_G(\Theta_{AE}, \Theta_F, \Theta_D)}{\partial \Theta_F} + \lambda_p \frac{\partial \mathcal{L}_{\text{partial}}(\Theta_{AE}, \Theta_F)}{\partial \Theta_F} + \lambda_c \frac{\partial \mathcal{L}_{\text{cycle}}(\Theta_{AE}, \Theta_F)}{\partial \Theta_F}], \quad (12)$$

where $\{\lambda_g, \lambda_c, \lambda_p\}$ are weight factors. Note that although \mathcal{L}_G and $\mathcal{L}_{\text{partial}}$ involve the parameter Θ_{AE} , we fix the parameter Θ_{AE} when training \mathcal{L}_G and $\mathcal{L}_{\text{partial}}$. The reason is that both \mathcal{L}_G and $\mathcal{L}_{\text{partial}}$ are constraints for the transformation process, while the two auto-encoders aim to learn a latent representation space instead of transferring features between complete and incomplete domains. The weight factors are fixed to $\lambda_g=1$, $\lambda_c=0.01$ and $\lambda_p=1$ in our experiments. Finally, the discriminators D_X and D_Y are regularized by the discriminator loss \mathcal{L}_D

$$\Theta_D \leftarrow \Theta_D - \gamma \frac{\partial \mathcal{L}_D(\Theta_{AE}, \Theta_F, \Theta_D)}{\partial \Theta_D}. \quad (13)$$

The pseudo code for training is given in **Algorithm 1**.

Algorithm 1 Pseudo code for training Cycle4Completion. The critic step n_D is fixed to 3 during training.

```

1: while model has not converged do
2:   Update  $\Theta_{AE}$  following Eq.11
3:   for  $t = 0, \dots, n_D$  do
4:     Update  $\Theta_D$  following Eq.13
5:   end for
6:   Update  $\Theta_F$  following Eq.12
7: end while

```

4. Experiments

4.1. Evaluation on ShapeNet Dataset

Dataset. Following the previous studies [3, 2], we evaluate our methods on 3D-EPN dataset [3], in order to fairly compare Cycle4Completion with the previous unpaired point cloud completion methods. For each 3D object, 8 partial point clouds are generated by back-projecting 2.5D depth images from 8 views into 3D. We uniformly sample only 2,048 points on the mesh surfaces for both the complete and partial shapes.

Quantitative and qualitative evaluation. We use the per point Chamfer distance as the evaluation metric. In Table 1, we compare Cycle4Completion with some state-of-the-art supervised and unpaired point cloud completion methods. Since the training and testing split of Pcl2Pcl and 3D-EPN is different at ShapeNet dataset, we report our results on both of the two splittings for fair comparison. In Table 1, the *Cycle4Completion* is the results of 3D-EPN splitting

Table 1. Point cloud completion comparison on ShapeNet dataset in terms of per point Chamfer distance $\times 10^4$ (lower is better).

Methods	Supervised	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat
3D-EPN [3]	Yes	29.1	60.0	27.0	24.0	16.0	38.0	45.0	14.0	9.0
FoldingNet [43]	Yes	9.2	2.4	8.5	7.2	10.3	14.1	9.1	13.6	8.8
PCN [46]	Yes	7.6	2.0	8.0	5.0	9.0	13.0	8.0	10.0	6.0
TopNet [33]	Yes	8.4	2.5	8.8	5.9	9.3	12.0	8.4	13.5	7.1
SA-Net [40]	Yes	7.7	2.2	9.1	5.6	8.9	10.0	7.8	9.9	7.2
AE (baseline)[2]	No	25.4	4.0	37.0	19.0	31.0	26.0	30.0	44.0	12.0
Pcl2Pcl [2]	No	17.4	4.0	19.0	10.0	20.0	23.0	26.0	26.0	11.0
Cycle4Completion (Ours)	No	14.1	3.1	10.9	7.5	14.6	16.7	26.7	24.5	9.1
Cycle4Completion* (Ours)	No	14.3	3.7	12.6	8.1	14.6	18.2	26.2	22.5	8.7

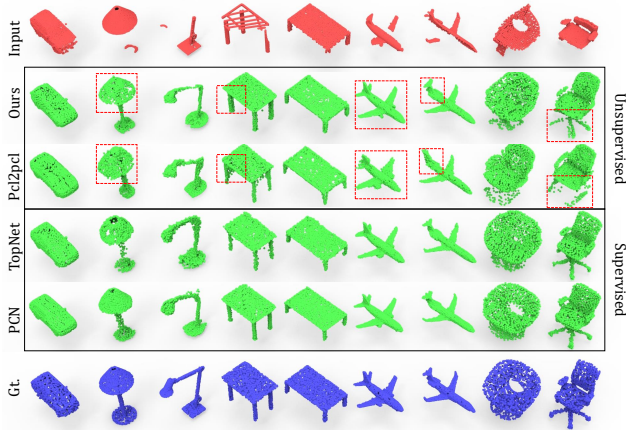


Figure 4. Visual comparison with the state-of-the-art completion methods on ShapeNet dataset.

and the *Cycle4Completion** is the results of Pcl2Pcl splitting. Moreover, we also quote the results of the baseline auto-encoder from [2] for comparison. The experimental results show that our method achieves the best completion performance on all categories compared with the unpaired counterpart method Pcl2Pcl [2]. And even comparing with the supervised methods, our Cycle4Completion still outperforms 3D-EPN [3] and yields a comparable results to PCN [46] and TopNet [33]. In Figure 4, we show the visualization results of point cloud completion using Cycle4Completion and compare it with other methods, from which we can find that our model predicts the complete shapes with higher accuracy than the unpaired Pcl2Pcl, especially on the regions highlighted by red rectangles. And the completion quality of our method is also comparable to the results of supervised methods.

4.2. Evaluation on KITTI Dataset

We supplement the following qualitative results on the KITTI dataset, which contains car objects in real-world auto-navigation dataset scanned by LIDAR sensor in streets. The Cycle4Completion is first trained on ShapeNet dataset under car category, and then the trained Cycle4Completion is directly used to predict complete shapes on the KITTI dataset without any further fine-tuning

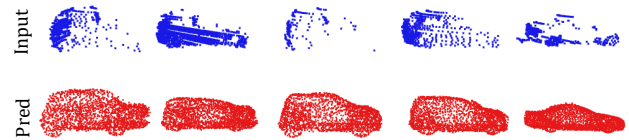


Figure 5. Visualizing the completion results of more similar car objects acquired from the KITTI dataset.

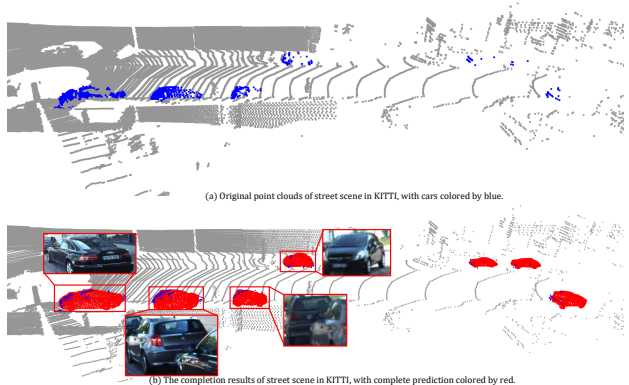


Figure 6. Visualizing the KITTI dataset with multiple completion cars.

In Figure 5, we show the original incomplete point clouds of cars (highlighted with blue) directly obtained by LIDAR sensor, and the complete shape (highlighted with red) predicted by our Cycle4Completion. In Figure 6, we further show the completion results integrated into the streets scene. Although there is no ground truth (i.e. the complete shapes of cars) for the KITTI dataset, we can still qualitatively find that Cycle4Completion predicts complete cars very robustly on the KITTI dataset, even our model is only trained on the ShapeNet dataset. This experiment shows that our Cycle4Completion model trained on ShapeNet dataset can achieve good completion results for more similar objects in the real-world scenario data.

4.3. Model Analysis

For clarity, we typically analyze the performance of Cycle4Completion on four categories, i.e. plane, cabinet, car and chair.

Visual analysis of complete cycle transformation. We

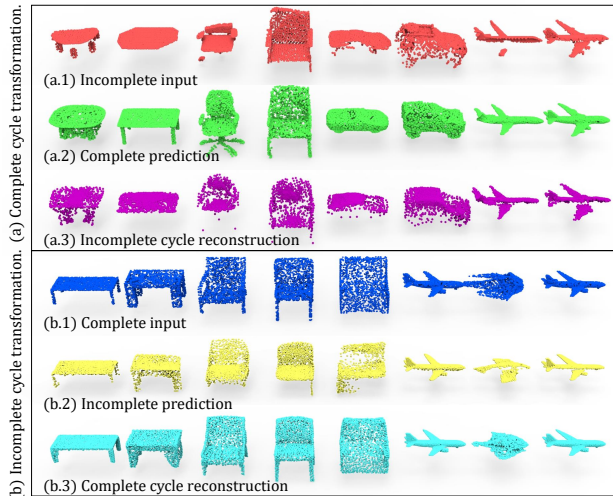


Figure 7. Visualization of the generated shapes in each step of cycle transformation.

visualize the results of incomplete-cycle in Figure 7(a), and the results of complete-cycle in Figure 7(b). For incomplete-cycle, the input to the network is the incomplete shape, as shown in Figure 7(a.1). The complete prediction in Figure 7(a.2) demonstrates the effectiveness of Cycle4Completion to produce complete shapes from incomplete input, and the comparison between Figure 7(a.1) and Figure 7(a.3) proves that Cycle4Completion successfully learns to keep shape consistency throughout the whole cycle transformation. Similar conclusions can also be drawn from Figure 7(b) for the complete-cycle.

Visual analysis of incompleteness quality. In Figure 8, we visually evaluate the quality of incomplete shapes, which are generated by Cycle4Completion from the complete input on a specific category of chair. The visual comparison between the incomplete prediction in Figure 8(a) and the real incomplete shapes in Figure 8(b) proves that Cycle4Completion successfully learns the geometric correspondence from the complete shapes to the incomplete ones. The similar pattern of incompleteness to the real incomplete shapes justifies the good 3D shape understanding ability of our model.

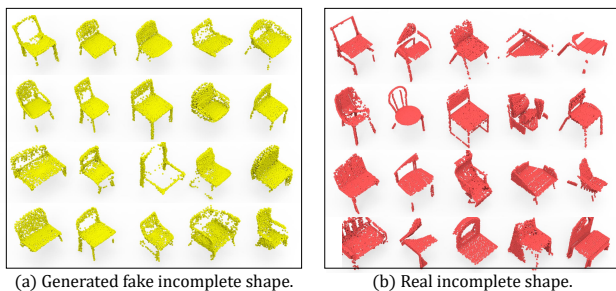
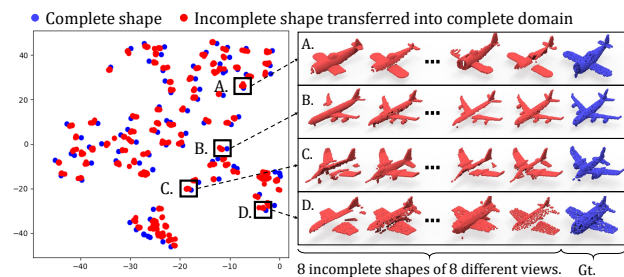


Figure 8. Visualization comparison of predicted incomplete shapes in (a) with the real incomplete shapes in (b).

Visual analysis of latent space. In Figure 9(a), we use t-

SNE[25] to visualize the latent features of complete shape and the ones of incomplete shapes that are transferred from incomplete domain into the complete domain. The red points stand for the latent features of incomplete shapes, and the blue points stand for the complete ones. Note that the incomplete shape is generated from partial views of complete ones, and we have 8 partial shapes generated from 8 different views of each complete shape. In Figure 9(a), we can find that red and blue points are arranged in a *paired pattern*, and from Figure 9(a) we can find that in each local area highlighted by black rectangles, there is a pair of one complete shape and its 8 partial incomplete shapes, which are exactly the geometric corresponding pattern between complete and incomplete shapes in the dataset. The visualization of latent space shows the effectiveness of Cycle4Completion to establish a well arranged latent space, and the ability to capture the shape correspondence between the complete and incomplete ones.



(a) Visualization of complete domain. (b) Visualization of shapes in marked areas.

Figure 9. Visualization of the latent representation space in complete domain. We randomly choose four areas in (a) and visualize the shape represented by these points in (b).

Effect of each module to our model. In order to analyze the effect of each module to our model, we develop three variations by removing element from Cycle4Completion including: (1) *w/o Partial* is the variation that removes the partial matching loss; (2) *w/o GAN* is the variation that removes the discriminator and its corresponding adversarial loss; (3) *w/o Cycle* is the variation that removes the cycle matching loss; (4) *w/o Coding* is the variation that removes all the missing region codes and the corresponding code matching loss. The results are shown in Table 2, where the *Full model* represents the original version of Cycle4Completion. In Table 2, the Full Model achieves the best completion performance, which proves the contributions of each part to the performance of Cycle4Completion. Moreover, *w/o Partial* variation yields the worst completion performance. This is because the model loses its supervision for learning the shape consistency when transferring representations from incomplete domain to complete one.

Effect of input point number. In order to further evaluate the performance of Cycle4Completion on more sparse input, we evaluate Cycle4Completion using the input of partial point clouds with different resolutions. Specifically,

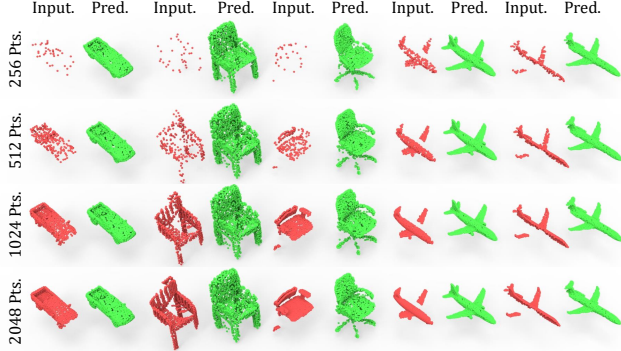


Figure 10. Visualization of completion performance under different input point numbers.

Table 2. The effect of each part (per point $CD \times 10^4$).

Methods	Average	Plane	Cabinet	Car	Chair
w/o Partial	23.7	15.6	27.8	14.8	36.6
w/o GAN	12.8	4.7	18.4	9.1	19.0
w/o Cycle	10.4	3.6	12.5	8.9	15.8
w/o Coding	9.4	3.2	11.8	7.7	14.8
Full Model	9.1	3.1	10.9	7.5	14.6

we keep the number of 2048 points on the output complete shape unchanged, and evaluate the performance of Cycle4Completion on the input point clouds with resolutions ranging from 256 to 2048. The quantitative completion results are given in Table 3, and the visualization results are shown in Figure 10. Both quantitative and qualitative results demonstrate a robust performance of Cycle4Completion on various input resolutions.

Table 3. The effect of input point number ($CD \times 10^4$).

#Points	Average	Plane	Cabinet	Car	Chair
256	14.4	3.3	15.9	9.9	28.4
512	10.0	3.2	12.1	7.9	16.9
1024	9.6	3.2	11.9	7.9	15.5
2048	9.1	3.1	10.9	7.5	14.6

Effect of different training strategies. In our model, the generator loss \mathcal{L}_G , cycle matching loss \mathcal{L}_{cycle} and partial matching loss $\mathcal{L}_{partial}$ involves multiple parameter sets. We selectively update some of the parameter sets while remain the others unchanged when training these losses. To evaluate the effectiveness of other potential training strategies, we develop the variation of (a) $\partial\mathcal{L}_G/\partial(\Theta_{AE}, \Theta_F)$ which updates both Θ_{AE} and Θ_F when training generator loss. Similarly, we also develop the variations of (b) $\partial\mathcal{L}_{partial}/\partial(\Theta_{AE}, \Theta_F)$ and (c) $\partial\mathcal{L}_{cycle}/\partial(\Theta_{AE}, \Theta_F)$ and report the results in Table 4. We observe a severe mode collapse in both variations (a) and (b), which is caused by training the auto-encoder cross different domain (both $\mathcal{L}_{partial}$ and \mathcal{L}_{AE} involve transformation from one domain to the other). In contrast, \mathcal{L}_{cycle} is a regularization considering one domain, and only involves reconstruction to input.

Table 4. The effect of training strategies ($CD \times 10^4$).

Strategies	Average	Plane	Cabinet	Car	Chair
$\partial\mathcal{L}_G/\partial(\Theta_{AE}, \Theta_F)$	collapsed	-	-	-	-
$\partial\mathcal{L}_{partial}/\partial(\Theta_{AE}, \Theta_F)$	collapsed	-	-	-	-
$\partial\mathcal{L}_{cycle}/\partial(\Theta_{AE}, \Theta_F)$	10.1	3.3	12.5	9.4	15.1
Original	9.1	3.1	10.9	7.5	14.6

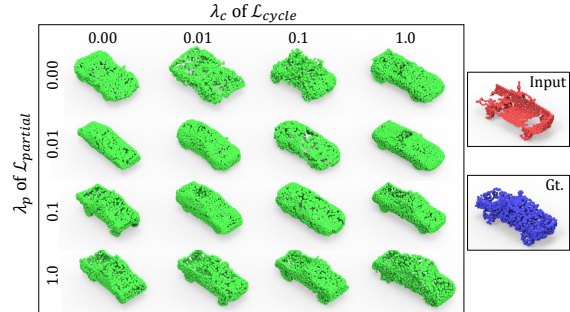


Figure 11. Visualization comparison of completion performance under different weight parameters λ_p and λ_c .

Effect of λ_c and λ_p . The cycle matching loss tends to keep shape consistent throughout the whole cycle transformation, while the partial matching loss only keeps a single side consistency. Different ratio between λ_c and λ_p will result in different preference of model to establish the shape consistency. In Table 5, we quantitatively analyze the effect of weight factors λ_c and λ_p to our model on the specific car class, and in Figure 11, we visually evaluate the corresponding completion performance. Since the completion task is a single-side transformation, a larger weight of λ_p yields better performance. However, as shown in Table 2, totally removing partial matching loss ($\lambda_p=0$) will degrade the performance of our model.

Table 5. The effect of λ_p and λ_c ($CD \times 10^4$).

$\lambda_p \backslash \lambda_c$	0.0	10^{-2}	10^{-1}	1.0
0.0	18.8	17.1	16.9	14.8
10^{-2}	15.8	15.6	12.4	12.4
10^{-1}	9.8	10.2	9.8	9.4
1.0	8.9	7.5	8.1	9.3

5. Conclusions

We propose the Cycle4Completion for unpaired point cloud completion task. Our model successfully captures the bidirectional geometric correspondence between incomplete and complete shapes, which enables the learning of point cloud completion without the paired complete shapes. Our model effectively learns to generate fake incomplete shapes to guide the completion network. The proposed Cycle4Completion is evaluated on the widely used ShapeNet dataset, and the experimental results demonstrate the state-of-the-art performance compared with other unpaired completion methods.

References

- [1] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, Andrei Sharf, and Claudio Silva. State of the art in surface reconstruction from point clouds. In *Proceedings of the Conference of the European Association for Computer Graphics*, volume 1, pages 161–185, 2014. **2**
- [2] 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, 3, 4, 5, 6**
- [3] 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. **3, 5, 6**
- [4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in neural information processing systems*, pages 5767–5777, 2017. **5**
- [5] Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 85–93, 2017. **3**
- [6] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning (ICML)*, 2020. **3**
- [7] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *International Joint Conference on Artificial Intelligence*, 2019. **1**
- [8] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and C.L. Philip Chen. 3D2SeqViews: Aggregating sequential views for 3D global feature learning by CNN with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019. **3**
- [9] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. Reconstructing 3D shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 2020. **3**
- [10] Zhizhong Han, Guanhai Qiao, Yu-Shen Liu, and Matthias Zwicker. SeqXY2SeqZ: Structure learning for 3D shapes by sequentially predicting 1D occupancy segments from 2D coordinates. In *European Conference on Computer Vision (ECCV)*, 2020. **3**
- [11] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *33rd AAAI Conference on Artificial Intelligence*, 2019. **1**
- [12] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Junwei Han, Matthias Zwicker, and C.L. Philip Chen. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by RNN with attention. *IEEE Transactions on Image Processing*, 28(2):658–672, 2019. **1, 3**
- [13] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-Angle Point Cloud-VAE: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10442–10451, 2019. **3**
- [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. In *International Joint Conference on Artificial Intelligence*, 2019. **1**
- [15] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. In *Proceedings of International Conference on Computer Vision*, 2019. **3**
- [16] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3D shape completion with multi-view consistent inference. In *AAAI*, 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**
- [18] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. **3**
- [19] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics*, 31(4):55, 2012. **2**
- [20] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *AAAI*, 2020. **1**
- [21] 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. In *The 14th International Conference on Geometric Modeling and Processing*, 2020. **1**
- [22] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *33rd AAAI Conference on Artificial Intelligence*, 2019. **1, 3**
- [23] 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**
- [24] 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. ACM, 2019. **1**
- [25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. **7**

- [26] Andelo Martinovic and Luc Van Gool. Bayesian grammar learning for inverse procedural modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 201–208. IEEE, 2013. 2
- [27] 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*, 2017. 1
- [28] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. RL-GAN-Net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5898–5907, 2019. 3
- [29] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. An interactive approach to semantic modeling of indoor scenes with an RGB-D camera. *ACM Transactions on Graphics*, 31(6):136, 2012. 2
- [30] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics*, 31(6):180, 2012. 2, 3
- [31] David Stutz and Andreas Geiger. Learning 3D shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 3
- [32] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 34(6):175, 2015. 2, 3
- [33] Lyne P Tchampi, Vineet Kosaraju, Hamid Rezaatofghi, Ian Reid, and Silvio Savarese. TopNet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 3, 6
- [34] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A field model for repairing 3D shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5684, 2016. 2
- [35] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3
- [36] Hu Wei, Fu Zeqing, and Guo Zongming. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Transactions on Image Processing*, 28(8):4087–4100, 2019. 2
- [37] Xin Wen, 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3
- [38] Xin Wen, Zhizhong Han, Xinhai Liu, and Yu-Shen Liu. Point2SpatialCapsule: Aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. *IEEE Transactions on Image Processing*, 29:8855–8869, 2020. 3
- [39] Xin Wen, Zhizhong Han, Geunhyuk Youk, and Yu-Shen Liu. CF-SIS: Semantic-Instance segmentation of 3D point clouds by context fusion with self-attention. In *ACM International Conference on Multimedia (ACM MM)*, 2020. 1
- [40] 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 Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3, 6
- [41] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *ECCV*, 2020. 3
- [42] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. GRNet: Gridding residual network for dense point cloud completion. In *ECCV*, 2020. 1
- [43] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 6
- [44] 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
- [45] Jun Yuan, Changjian Chen, Weikai Yang, Mengchen Liu, Jiazhi Xia, and Shixia Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1), 2021. 2
- [46] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1, 6
- [47] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *ECCV*, 2020. 3
- [48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3