Symmetric Shape-Preserving Autoencoder for Unsupervised Real Scene Point Cloud Completion

Changfeng Ma[1], Yinuo Chen[1], Pengxiao Guo[1], Jie Guo[1], Chongjun Wang[1], Yanwen Guo[1*]
[1]Nanjing University, Nanjing, China
{changfengma,yinuochen,pxguo}@smail.nju.edu.cn
{guojie,chjwang,ywguo}@nju.edu.cn

Abstract

*Unsupervised completion of real scene objects is of vital importance but still remains extremely challenging in preserving input shapes, predicting accurate results, and adapting to multi-category data. To solve these problems, we propose in this paper an **U**nsupervised **S**ymmetric **S**hape-**P**reserving **A**utoencoding Network, termed **USSPA**, to predict complete point clouds of objects from real scenes. One of our main observations is that many natural and man-made objects exhibit significant symmetries. To accommodate this, we devise a symmetry learning module to learn from those objects and to preserve structural symmetries. Starting from an initial coarse predictor, our autoencoder refines the complete shape with a carefully designed upsampling refinement module. Besides the discriminative process on the latent space, the discriminators of our USSPA also take predicted point clouds as direct guidance, enabling more detailed shape prediction. Clearly different from previous methods which train each category separately, our USSPA can be adapted to the training of multi-category data in one pass through a classifier-guided discriminator, with consistent performance on single category. For more accurate evaluation, we contribute to the community a real scene dataset with paired CAD models as ground truth. Extensive experiments and comparisons demonstrate our superiority and generalization and show that our method achieves state-of-the-art performance on unsupervised completion of real scene objects.*

1. Introduction

As the standard outputs of 3D scanners [12, 32], point clouds are becoming more and more popular [9] which are also the basic data structure in 3D geometry processing [4, 5, 13]. Complete point clouds are hard to obtain due to the nature of the scanning process and object occlusion [35]. Due to the defects of incomplete point clouds on downstream applications such as reconstruction [10], recent works [17, 19, 22, 23, 26, 30, 33, 35] pay more attention to point cloud completion which relies on paired artificial complete point clouds for supervised training to complete partial point clouds. However, these supervised works are difficult to apply in practice because of the great gap between artificial data and real scene data and the inaccessibility to the ground truth of real scene data. Therefore, it is important to complete partial point clouds from real scene in an unsupervised way.

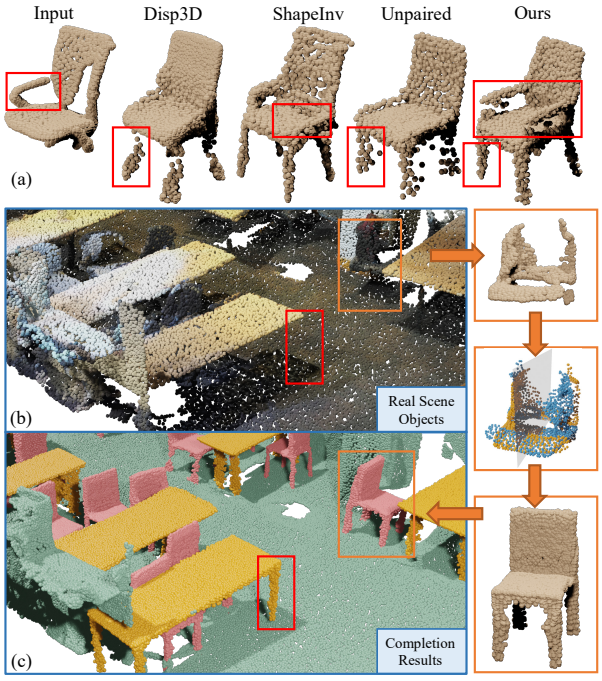

Figure 1. Visual comparison of predicted results on real scene data by our USSPA and other works (top) and our complete result on a whole real scene (bottom). (a) shows an example of a real scene partial point cloud of a chair and the complete predictions by Disp3D [23], ShapeInv [36], Unpaired [32] and our method. As shown, our prediction result is more accurate and uniform according to the input, which contains complete arms and legs. (b) and (c) show the original point cloud of a real scene and the complete results of all the objects in this scene.

Recent unsupervised works [24, 32, 36] only require real

*Corresponding author.

scene partial point clouds and artificial CAD models for unpaired completion utilizing GANs [8] as their fundamental frameworks, most of which need pre-training on artificial data. The main ideas are to transform latent codes from the space of real scene partial data to the space of artificial complete data and then employ the decoder trained on artificial data to predict the complete point cloud. Essentially, these methods make the predictions whose distributions are consistent with the artificial models. Most of them, however, just extract a global feature from the partial input without fully exploiting its geometry information, leading to the prediction severely deviated from the input. And such information actually provides vital clues and constraints for completion. Furthermore, prediction results by these methods usually lack enough geometric details due to the absence of an explicit discriminative process on point clouds. These domain transforming methods are also hard to adapt to multi-category data or other datasets.

In this paper, we present an unsupervised symmetric shape-preserving autoencoding network, termed USSPA, for the completion of real scene objects, as shown in Figure 2, which is a GAN-based end-to-end network without the requirement of pre-trained weights. Different from previous domain transforming methods which cannot fully leverage existing incomplete models, we argue that the existing partial scanning, which also provides vital clues and constraints for the prediction of the missing part, should be preserved to some extent. To this end, we exploit the symmetries shown in many natural or man-made objects and devise a novel symmetry learning module to generate symmetrical point clouds of existing parts by predicting the symmetric planes. This enables our network to preserve the shapes of input symmetrically, intrinsically facilitating structure completion, as shown in Figure 1. For those parts that can not be directly inferred from inputs, we employ an initial coarse module for an initial prediction first. Starting from the initial guess, we specifically design a refinement autoencoder with an upsampling refinement module for detailed refinement and the local feature grouping for extracting local information, to learn detailed structures of artificial data through the autoencoding process. Benefiting from this, our final prediction is accurate, uniform, and symmetric shape-preserving. Besides the indirect guidance of the feature discriminator on latent space, our point discriminator takes predicted point clouds as direct guidance for generating more accurate shapes. Compared with previous methods which train each category separately, our method can classify the partial point clouds simultaneously through a classifier-guided discriminator when adapted to multi-category data, with consistent performance on the single category.

To measure the performance of unsupervised completion quantitatively, we build a dataset from ScanNet [5] and ShapeNet [2] utilizing the annotations of Scan2CAD [1]. Our dataset contains real scene partial point clouds and paired ground truths that are only used for evaluation in our experiments. Extensive comparisons against previous works on this dataset and the public PCN Dataset [35] show the superiority and generalization of our method which achieves state-of-the-art performance on unsupervised completion of real scene objects.

Our main contributions are as follows.

- We propose a novel USSPA for unsupervised real scene point cloud completion whose prediction is accurate, uniform and symmetric shape-preserving. Clearly different from previous works training each category separately, our USSPA can be adapted to the training of multi-category data in one pass by classifying the input simultaneously.
- We propose a novel symmetry learning module and a novel refinement autoencoder. The symmetry learning module preserves input shapes by generating symmetrical point clouds, and the refinement autoencoder learns the detailed information from artificial data to refine the initial guess by an autoencoding process.
- We propose a new evaluation method for obtaining paired ground truths and partial data from artificial and real scene datasets using alignment information, which can be used to more accurately evaluate unsupervised completion of real scene objects.

2. Related Work

2.1. Supervised Point Cloud Completion

To infer the complete point cloud from an incomplete input, many deep learning methods were introduced in recent years, most of which are supervised. At first, the 3D data were voxelized into occupancy grids or distance fields before fed into convolutional networks [15, 16, 20]. To avoid loss of details and huge memory cost brought by these data representations, Yuan et al. [35] propose PCN which directly operates on raw point clouds. Giancola et al. [7] also leverage a new method for 3D shape completion which directly works on LiDAR point clouds. Later works [11, 19, 22] pay much attention to details refinement and denoise. Disp3D [23] investigates grouping local features to bring improvement to point cloud completion. Taking structures and topology information into consideration, SA-Net [25], GRNet [31], PoinTr [34] and LAKe-Net [17] succeed to utilize the detailed geometry information from input to generate more reasonable complete point clouds. Other works like designing new metrics [28] and combining information of single-view images [37] also make contributions to the point cloud completion task. However, these works are all supervised that are not suitable for practice since ground truths are in lack.

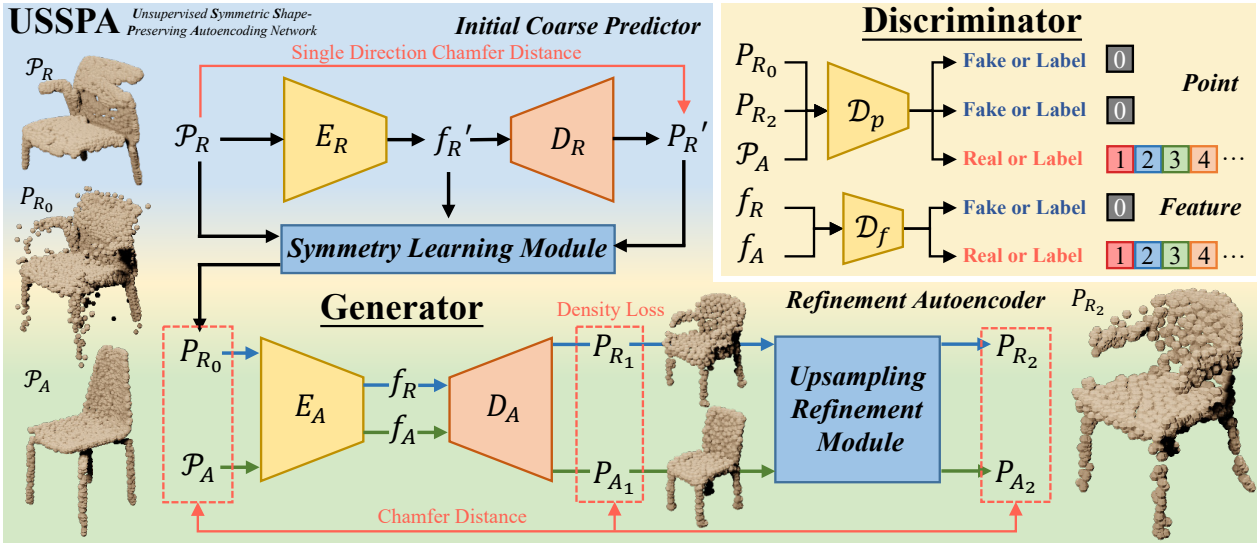

Figure 2. The overall architecture of our USSPA. Our network has a generator composed of an initial coarse predictor (blue) and a refinement autoencoder (green), and the point and feature discriminators (yellow). The red lines and boxes represent the employed losses. The blue and green lines (bottom) denote the forward processes having the same network architectures with sharing weights.

2.2. Unsupervised Point Cloud Completion

Due to the absence of suitable paired training data for supervised models, some unsupervised models for point cloud completion have emerged in recent years. Unpaired [32] first introduces the GAN to learn the transformation from the latent space of partial point cloud to the latent space of complete point cloud, and then decode the transformed latent code to predict. Zhang et al. [36] employ optimization-based GAN inversion [29] and directly find the complete prediction by optimizing the latent code such that the degradation of the complete point cloud matches the input partial point cloud, which requires fine-tuning the network for each input data. Cycle4 [24] proposes two simultaneous cycle transformations between the latent spaces of complete point cloud and partial ones, facilitating the network's understanding of 3D shapes by building a bidirectional mapping. Wu et al. [27] propose the first multimodal shape completion method that completes the partial shape via conditional generative modeling, without requiring paired training data. However, the application of pre-trained GANs makes their predictions lack details and difficult to preserve the shape features of the input. For evaluation, Unpaired [32] employs classification accuracy of the complete results which is hard to measure detailed structures. Thus, a more accurate evaluation method is required.

3. Method

3.1. Network Architecture

Given the partial point cloud $\mathcal{P}_R \in \mathbb{M}_{n_0 \times 3}$ from real scene data with distribution p_r and the complete point cloud $\mathcal{P}_A \in \mathbb{M}_{n_0 \times 3}$ from artificial data with distribution p_a, the goal of unsupervised real scene point cloud completion is to predict the complete point cloud $Com(\mathcal{P}_R)$ of $\mathcal{P}_R$ such that $Com(\mathcal{P}_R) \sim p_a$ and $\mathcal{P}_R \subset Com(\mathcal{P}_R)$. This means that the prediction must be similar to the artificial data and meanwhile retain the shapes of the input partial point cloud.

We carefully design several modules of our generator $\mathcal{G}$ and discriminators, $\mathcal{D}_p$ and $\mathcal{D}_f$, for unsupervised completion on real scene data through generative adversarial learning [8] as shown in Figures 2 and 3. As we mentioned above, many natural and man-made objects exhibit significant symmetries. To fully leverage these symmetries and preserve the shape of existing partial input $\mathcal{P}_R$, we devise a novel symmetry learning module that predicts the symmetric plane and the symmetrical point cloud of $\mathcal{P}_R$ for further refinement. Our refinement autoencoder learns the detailed information of artificial data and refines the details of the initial, but coarse prediction P_{R_0} for uniform and accurate prediction. With the local feature grouping operation, our elaborate upsampling refinement module can utilize the local features for more detailed results. Different from previous works focusing on the discriminative process of latent space, our point discriminator $\mathcal{D}_p$ also takes point clouds as input for direct guidance.

Initial Coarse Predictor. We first employ an initial coarse predictor for shape-preserving and preliminary prediction. The encoder E_R takes the real scene partial point cloud $\mathcal{P}_R$ as input and encodes the global feature $f'_R = E_R(\mathcal{P}_R) \in \mathbb{M}_{1 \times c_0}$. f'_R is then taken by the decoder D_R for the prediction of initial, but coarse complete result $P'_R = D_R(f'_R) \in \mathbb{M}_{n_1 \times 3}$. Encoder-decoder architecture is hard to retain shape features on the completion task since there is no explicit complete ground truth. To this end, we devise the **symmetry learning module** to accommodate symmetrical shape features widely existed in man-made and natural objects, as shown in Figure 3. The decoder D_{sl} pre-

dicts symmetric arguments $\boldsymbol{A} = D_{sl}(f'_R) = [x_A, y_A, z_A]^T$ and then we introduce *symmetry operation* to calculate the symmetrical point cloud P_{sym} of $\mathcal{P}_R$ by :

$$P_{sym} = \boldsymbol{p} - 2\frac{\boldsymbol{A}\cdot\boldsymbol{p}}{||\boldsymbol{A}||^2}\boldsymbol{A}, \tag{1}$$

where $\boldsymbol{p} = [x, y, z]^T \in \mathcal{P}_R$, $\cdot$ represents the inner product and $||\boldsymbol{x}||$ denotes the module of the vector $\boldsymbol{x}$. Without the loss of generality, we can safely assume that all the objects are axis-aligned. Under such an assumption, the symmetric plane is perpendicular to the xz-plane and zero-crossing. Thus we set $y_A = 0$. Then P_{sym}, $\mathcal{P}_R$ and P'_R are fused and resampled through *FPS (farthest point sampling)* to get P_{R_0} which is the complete and shape-preserving result.

Refinement Autoencoder. Noisy $\mathcal{P}_R$ and the fusion operation may cause noisy P_{R_0} with uneven point distribution. The initial coarse predictor or the point discriminator only predicts or judges through the global feature, which makes P_{R_0} lack details. Thus, P_{R_0} requires denoising and further refinement on details and point distribution. To this end, we employ the refinement autoencoder with a carefully designed upsampling refinement module. As shown in Figure 2, the blue and green lines of the refinement autoencoder represent the same forward process having the same network architectures with sharing weights. In this way, our autoencoder extracts the detailed information of artificial data and refines the details of P_{R_0} for more accurate and uniform prediction.

The encoder and decoder E_A, D_A generate the skeleton point cloud $P_{R_1/A_1} \in \mathbb{M}_{n_2\times 3}$ of input point cloud. We set n_2 to a small number for easier prediction of P_{R_1/A_1}. And the density loss is faster to calculate with fewer points. We then employ **upsampling refinement module** to upsample P_{R_1/A_1} and add details to the skeleton for final prediction P_{R_2/A_2}. As shown in Figure 3, we utilize wight-shared MLP [3] to obtain the point features and max-pooling operation on point features to obtain the global feature of the point cloud. We also concatenate the point and global features to fuse the information. Since wight-shared MLP only extracts the feature of one point, lacking the local information, we also employ the *local feature grouping* to get the local information around each point. For point p_0, $f_{group} \in \mathbb{M}_{n_2\times(c_1\cdot m)}$ are the features grouped from $f_{input} \in \mathbb{M}_{n_2\times c_1}$ corresponding to the points p that $||p - p_0|| \leq r$, where m is the feature number of each group. After applying max-pooling operation on m features of each group, $f_{local} \in \mathbb{M}_{n_2\times c_1}$ are concatenated with f_{input} to generate the output $f_{output} \in \mathbb{M}_{n_2\times 2c_1}$. We utilize a weight-shared MLP for *upsmapling* that takes fused features composed of P_{R_1/A_1}, local and global features, and predict k shifts for each point. Then, each point splits into k points by adding the predicted shifts to its coordinate whose result is $P_{R_2/A_2} \in \mathbb{M}_{kn_2\times 3}$.

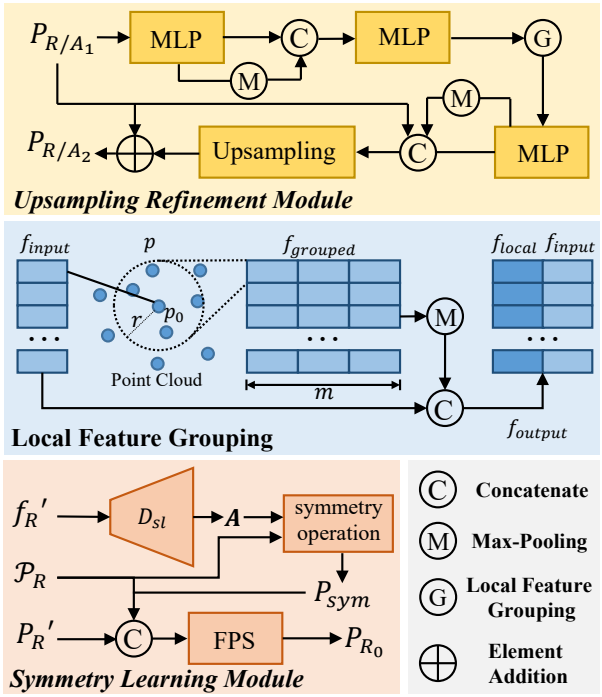

Figure 3. Detailed structures of our symmetry learning module (red), upsampling refinement module (yellow), and local feature sampling (blue).

Point and Feature Discriminator. We employ the point discriminator $\mathcal{D}_p$ to directly guide the generation of P_{R_0} and P_{R_2} comparing with the artificial complete data $\mathcal{P}_A$ for unsupervised completion. We also guide the generation of latent code of autoencoder through the feature discriminator $\mathcal{D}_f$ such that f_R and f_A have the same distribution for more accurate prediction. Our discriminators predict the probability that input point cloud P_i or feature f_i is real data. For multi-category data, we replace the probability-guided discriminator with a classifier-guided discriminator which outputs the category label of input rather than probability. We utilize label 0 to represent fake data and labels $l > 0$ to denote real data of different categories such as the chair, table, etc. For fake data, the discriminator tends to output label 0, while the generator tends to make the discriminator output label $l > 0$. And for real data with label l, the discriminator tends to output l correctly. In this way, our USSPA can unsupervisedly classify the real scene partial point cloud when completing multi-category data.

3.2. Optimization

The abbreviate optimization goal of our USSPA for unsupervised learning is:

$$\min_{\mathcal{G}} \max_{\mathcal{D}_p,\mathcal{D}_f} \mathop{\mathbb{E}}_{P_i\sim p_a} \log p + \mathop{\mathbb{E}}_{P_i\sim p_r} \log(1-p), \tag{2}$$

where p is the probability of real data. By playing the maxmin game with $\mathcal{D}_p$ and $\mathcal{D}_f$, our generator $\mathcal{G}$ tends to predict point clouds similar to artificial data. We employ

Table 1. Point cloud completion comparison on our dataset in terms of L1 Chamfer Distance $cd^{l1} \times 10^2$ (lower is better).All the methods are trained with our dataset and the unsupervised methods including ours are trained on single category data. Boldface denotes the best among unsupervised methods in Tables 1, 2 and 3.

	Method	AVG	chair	table	trash bin	TV	cabinet	bookshelf	sofa	lamp	bed	tub
Sup.	PoinTr [34]	14.37	13.65	12.52	15.26	12.69	17.32	13.99	12.36	17.05	15.13	13.77
	Disp3D [23]	7.78	6.24	8.20	7.12	7.12	10.36	6.94	5.60	14.03	6.90	5.32
	TopNet [19]	7.07	6.39	5.79	7.40	6.26	8.37	7.02	5.94	8.50	7.81	7.25
Unsup. OptBased	ShapeInv [36]	21.39	17.97	17.28	33.51	15.69	26.26	25.51	14.28	16.69	32.33	14.43
Unsup.	Unpaired [32]	10.47	8.41	**7.52**	12.08	**6.72**	17.45	9.95	6.92	19.36	10.04	**6.22**
	Cycle4 [24]	11.53	9.11	11.35	11.93	8.40	15.47	12.51	10.63	12.25	15.73	7.92
	Ours	**8.56**	**8.22**	7.68	**10.36**	7.66	**10.77**	**7.84**	**6.14**	**11.93**	**8.20**	6.75

Chamfer Distance(CD) [6] to measure the difference between point clouds. The $cd^{l2}_{P_1 \to P_2}$ indicates the single direction Chamfer Distance with L2-norm from point cloud P_1 to P_2:

$$cd^{l2}_{P_1 \to P_2} = \frac{1}{|P_1|} \sum_{\boldsymbol{x} \in P_1} \min_{\boldsymbol{y} \in P_2} ||\boldsymbol{x} - \boldsymbol{y}||^2, \tag{3}$$

which guides P_1 to be a part of P_2. The $cd^{l2}_{P_1 \leftrightarrow P_2}$ between P_1 and P_2 is:

$$cd^{l2}_{P_1 \leftrightarrow P_2} = cd^{l2}_{P_1 \to P_2} + cd^{l2}_{P_2 \to P_1}, \tag{4}$$

which guides P_1 and P_2 to be identical. As shown in Figure 2, to restrict the generation of P'_R, we employ single direction Chamfer Distance from $\mathcal{P}_R$ to P'_R. We also utilize Chamfer Distance between P_{R_1/A_1}, $P_{R_2/A_2}$and P_{R_0},$\mathcal{P}_A$ for supervision of autoencoder. For uniform prediction, we employ density loss by first calculating the local density of $\boldsymbol{x} \in P$:

$$ld(\boldsymbol{x}, P) = \frac{1}{k_d} \sum_{k=1}^{k_d} \min_{\boldsymbol{y} \in P}^{k} ||\boldsymbol{x} - \boldsymbol{y}||, \tag{5}$$

and then calculating the standard deviation for all $\boldsymbol{x}$:

$$dl(P) = \sqrt{\frac{1}{|P|} \sum_{\boldsymbol{x} \in P} \left(ld(\boldsymbol{x}, P) - \overline{ld(\boldsymbol{x}, P)}\right)^2}, \tag{6}$$

where $\min^{k}$ indicates the k-th-min element. The smaller the $ld(\boldsymbol{x}, P)$ is, the bigger the local density of $\boldsymbol{x}$ is.

In practice, we alternately optimize $\mathcal{G}$ with loss $\mathcal{L}_\mathcal{G}$:

$$\mathcal{L}_\mathcal{G} = \alpha_1 \mathcal{L}_{F \to R} + \alpha_2 \mathcal{L}_P + \alpha_3 \mathcal{L}_d, \tag{7}$$

where

$$\begin{aligned} \mathcal{L}_{F \to R} = -\big(&\log \mathcal{D}_p(P_{R_0}) + \log \mathcal{D}_p(P_{R_2}) \\ &+ \log \mathcal{D}_f(f_R)\big), \end{aligned} \tag{8}$$

$$\begin{aligned} \mathcal{L}_P = &cd^{l2}_{P_{R_1} \leftrightarrow P_{R_0}} + cd^{l2}_{P_{R_2} \leftrightarrow P_{R_0}} + cd^{l2}_{\mathcal{P}_R \to P'_R} \\ &+ cd^{l2}_{P_{A_1} \leftrightarrow \mathcal{P}_A} + cd^{l2}_{P_{A_2} \leftrightarrow \mathcal{P}_A}, \end{aligned} \tag{9}$$

$$\mathcal{L}_d = dl(P_{R_1}) + dl(P_{A_1}) \tag{10}$$

and $\mathcal{D}$ with loss $\mathcal{L}_\mathcal{D}$:

$$\mathcal{L}_\mathcal{D} = \alpha_4 \mathcal{L}_F + \alpha_5 \mathcal{L}_R, \tag{11}$$

where

$$\begin{aligned} \mathcal{L}_F = -\big(&\log[1 - \mathcal{D}_p(P_{R_0})] + \log[1 - \mathcal{D}_p(P_{R_2})] \\ &+ \log[1 - \mathcal{D}_f(f_R)]\big), \end{aligned} \tag{12}$$

$$\mathcal{L}_R = -\big(\log \mathcal{D}_p(\mathcal{P}_A) + \log \mathcal{D}_f(f_A)\big), \tag{13}$$

and here $\alpha_{1\sim5}$ are the weights used for balancing the influences between each term. And for multi-category data, we use cross-entropy to calculate the loss between predicted labels and ground truth labels.

4. Experiments

4.1. Dataset and Implementation Details

Different from the Unpaired [32] which utilizes a classifier to evaluate the completion results of real scene data, we build a new dataset from ScanNet [5], ShapeNet [2] and Scan2CAD [1], containing paired partial and complete point clouds for more accurate evaluation. Scan2CAD contains annotations that align the CAD models of ShapeNet to the real scenes of ScanNet. We first extract the CAD models from ShapeNet and partial point clouds from ScanNet which are paired according to the correspondences offered by Scan2CAD, and then uniformly sample the CAD models to get complete point clouds. Finally, we transform the paired partial and complete point clouds with the transform matrix given by Scan2CAD to normalize their face direction, scale, and position. We select ten categories including the chair, table, etc., to establish the dataset and randomly split the data into training, validation, and testing sets. In our experiments, these ground truths are only used for the evaluation of the complete results.

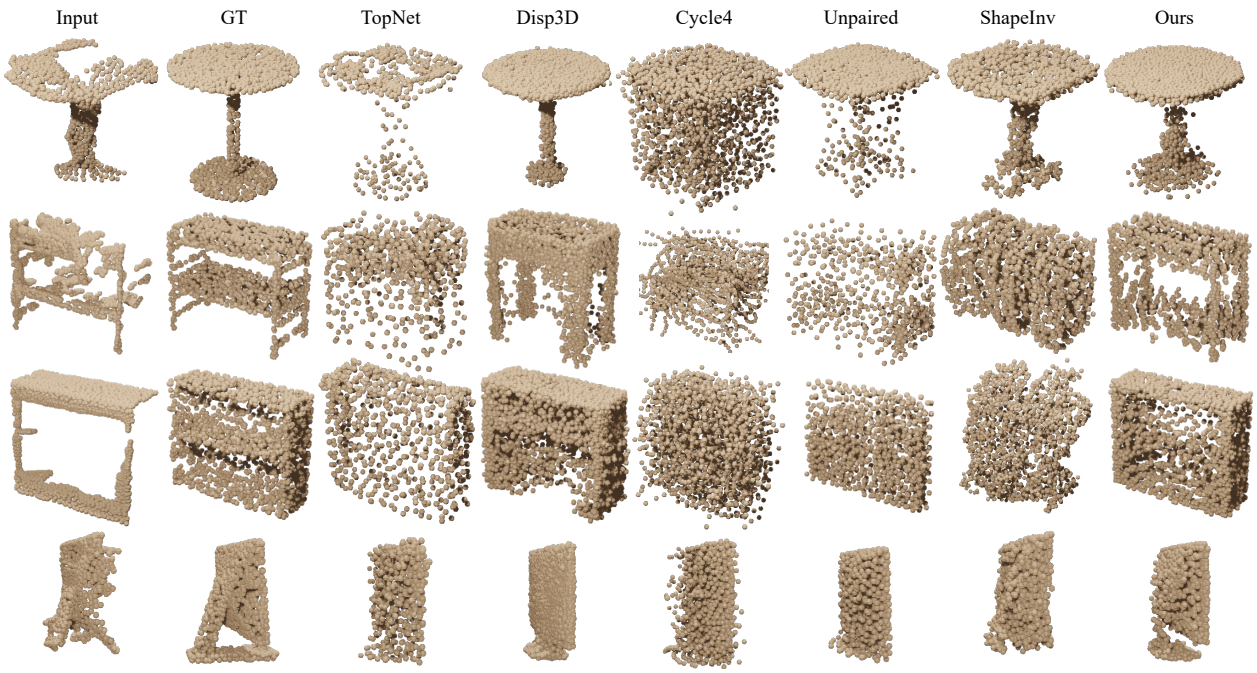

Figure 4. Visualization of completion results by supervised works, including TopNet [19], Disp3D [23], unsupervised works Cycle4 [24], Unpaired [32], optimization-based work ShapeInv [36] and our method. From top to bottom: table, bed, cabinet, and TV.

We utilize PyTorch to implement our USSPA. In Section 3.1, the point numbers $n_{0\sim2}$ are set to 2048, 512 and 512. The channel numbers $c_{0\sim1}$ are set to 512 and 256. The feature number m, the radius r of local feature grouping, and the split number of upsampling k are set to 32, 0.5, and 4. k_d of density loss in Section 3.2 is set to 16. The weights $\alpha_{1\sim5}$ of losses are set to 1, 100, 7.5, 0.5 and 0.5. We employ Adam optimizer whose learning rate is 1.0×10^{-5}. The batch size is set to 4, and the maximum epoch of training is 240 to 960 according to the size of the dataset. Training our USSPA takes about 20 hours with a GTX 2080Ti GPU. We will make our dataset and codes public in the future.

4.2. Comparison

We compare our USSPA against existing representative point cloud completion methods including supervised [19, 23, 34], unsupervised [24, 32] and optimization-based [36] methods on our dataset and the PCN Dataset [35]. The metrics used for comparison include pre-point L1 Chamfer Distance cd^{l1} [6] and F-Scores $F^{1\%}_{score}$, $F^{0.1\%}_{score}$ [18].

As shown in Table 1, we compare these methods on our dataset with 10 categories where all methods are trained on the training set and evaluated on the testing set. All the unsupervised methods are trained with single category data for each class. The average cd^{l1} of our USSPA is 18.2% lower than Unpaired and the average $F^{0.1\%}_{score}$ is 30% higher than Unpaired as shown in Table 2, indicating the superior performance of our method on completion of real scene objects. We also mix up all 10 categories as the multi-category data to train the methods for evaluation of the generalization ability. As shown in Table 2, our USSPA keeps the same performance by replacing the probability-guided discriminators with classifier-guided discriminators and has better generalization ability on multi-category than other unsupervised networks. Further comparisons show the generalization capability of our USSPA on different datasets. We train unsupervised models with ScanNet and then test them on the PCN Dataset which is a point cloud completion dataset built upon artificial data. As shown in Table 3, our method also performs better than other unsupervised methods, the average cd^{l1} of which is 15% lower than the SOTA method. Table 4 shows the classification accuracy of PN2 [14] and

Table 2. Comparison on our dataset trained with single category and multi-category data in terms of L1 Chamfer Distance $cd^{l1} \times 10^2$ (lower is better) and F-scores $F^{0.1\%}_{score} \times 10^2$, $F^{1\%}_{score} \times 10^2$ (higher is better). "Ours" and "Ours(classifier)" denote our method with probability-guided and classifier-guided discriminators, respectively.

Method	single category		multi-category		
	$\uparrow F^{0.1\%}_{score}$	$\uparrow F^{1\%}_{score}$	$\downarrow cd^{l1}$	$\uparrow F^{0.1\%}_{score}$	$\uparrow F^{1\%}_{score}$
PoinTr [34]	-	-	14.37	18.35	80.41
Disp3D [23]	-	-	7.78	30.29	78.26
TopNet [19]	-	-	7.07	12.33	80.37
ShapeInv [36]	15.58	66.53	19.35	16.98	69.66
Unpaired [32]	12.20	64.33	10.12	10.86	66.68
Cycle4 [24]	9.98	60.14	12.00	8.61	56.57
Ours	**17.49**	**73.41**	8.96	16.88	72.31
Ours(classifier)	-	-	**8.76**	**17.12**	**73.75**

Table 3. Point cloud completion comparison on PCN Dataset in terms of L1 Chamfer Distance $cd^{l1} \times 10^2$ (lower is better). All unsupervised models are trained with ScanNet.

Method	AVG	chair	table	cabinet	sofa	lamp
PoinTr [34]	5.49	5.61	5.68	6.08	5.67	4.44
Disp3D [23]	2.51	2.42	2.30	2.38	2.44	3.00
TopNet [19]	5.92	6.34	5.45	6.06	5.80	5.95
ShapeInv [36]	19.05	23.18	15.66	17.14	22.85	16.40
Unpaired [32]	14.87	**12.87**	**8.14**	14.30	18.23	20.82
Cycle4 [24]	17.60	14.25	15.73	21.06	21.54	**15.40**
Ours	**12.63**	13.52	9.66	**8.89**	**15.51**	15.57

Table 4. Point cloud classification in terms of accuracy. "AVG" denotes the average accuracy on 10 categories of our dataset. Only four categories are shown due to width limitation.

Method	AVG	chair	table	trash bin	sofa
PN2 [14]	58.6%	**91.0%**	**94.0%**	17.2%	48.9%
Ours	**69.8%**	**91.0%**	91.0%	**51.6%**	**93.3%**

our method. Our method can also unsupervisedly classify the real scene partial point cloud while completing. The accuracy of our USSPA is 11.2% higher than PN2 on average.

Figure 4 shows the qualitative results on 4 categories. The complete point clouds predicted by our USSPA exhibit more accurate shapes, such as the leg of the table and the division plate of the bookshelf, which are hard to generate even for supervised methods. Our method can successfully generate the circular top of the table. By comparision, other unsupervised works generate a square one. Our predictions are also more uniform with fewer noises. Benefiting from our shape-preserving method, our predicted TV retains the support part from the input point cloud. Our symmetry learning module also plays an important role in completing the pillars of the bed by copying the pillars from the input symmetrically. Visualization of our predicted point clouds shows the superiority of our methods in the generation of detailed completion on real scene data.

4.3. Ablation Study

Ablation study on network architecture. We conduct experiments on ablated models for evaluation of the necessity of important structures of our USSPA, including the symmetry learning module, the refinement autoencoder, the local feature grouping, the point and feature discriminators, the single direction Chamfer Distance, and the upsampling refinement module. As shown in Table 5, these modules are removed from our full network separately. We downsample the final prediction of our USSPA to 512 points for a fair comparison to the ablated model without the upsampling refinement module. We train and utilize $F_{score}^{0.1\%}$ to evaluate these ablated models with our dataset on the chair category, and the comparison indicates the necessity of these modules

Table 5. Ablation study in terms of $F_{score}^{0.1\%}$, where "-" represents removing a module from the full network respectively and "512" denotes the point number.

Model	$F_{score}^{0.1\%}$
- Symmetry Learning Module	16.43
- Refinement Autoencoder	17.08
- Local Feature Grouping	15.72
- Feature Discriminator	17.11
- Point Discriminator	15.68
- Single Direction Chamfer Distance	17.84
Ours (full)	**18.43**
- Upsampling Refinement (512)	5.60
Ours (full) (512)	**6.46**

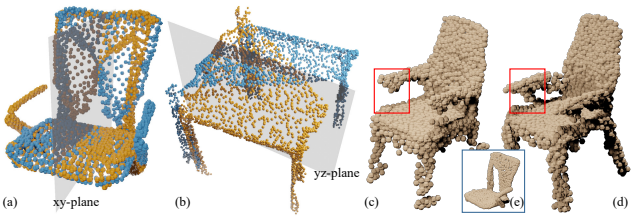

Figure 5. Visualization of the symmetric planes and predictions by USSPA without the symmetry learning module and full network. Blue and orange denote original and mirrored points, separately.

which play significant roles in shape-preserving, extracting, and utilizing detailed features, upsampling and refining for more accurate point cloud completion.

Symmetry learning module. Our symmetry learning module learns different symmetric planes according to the input point clouds as shown in Figures 5(a) and 5(b). The predicted symmetric plane is xy-plane for the chair and yz-plane for the table. Through different symmetric planes, our network can fully leverage the symmetrical structures of input point clouds, such as the arm of the chair and the legs of the table, which exist on only one side. Figure 5(c) shows the prediction P_{R_2} of our network without the symmetry learning module, and Figure 5(d) shows the result of our full network. Figure 5(e) shows the input point cloud whose left arm is missing while the right arm is complete. Without the symmetry learning module, the left arm of the prediction is incomplete as shown in the red box. And benefiting from such a module, our full network can learn the symmetric structures, and generate a complete left arm.

Density loss. We visualize the local density of points as shown in Figure 6. The warmer the color of the point is, the bigger its local density (described in equation 5) is. (a) is the prediction without the supervision of the density loss, and (b) is the prediction supervised by the density loss. Without the supervision of density loss, the network prefers generating many points in the same location as shown in the blue circle to minimize the Chamfer Distance, which makes the predicted point clouds contain fewer points describing

Table 6. Comparison of Unpaired and our USSPA in terms of density loss, where "ours (w/o dl)" indicates our USSPA without the supervision of the density loss. The results show that our predicted point clouds are more uniform.

	Unpaired	ours (w/o dl)	ours
Density Loss	0.0957	0.1238	**0.0149**

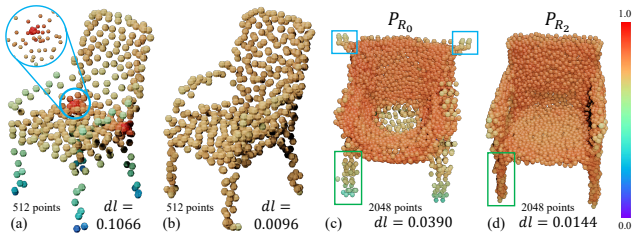

Figure 6. Visualization of the local density of point clouds, where the warmth of the color indicates the values of local densities. The warmer the color is, the bigger the local density is. (a) and (b) are the comparison of the ablation experiment on the density loss. (c) and (d) are the examples of P_{R_0} and P_{R_2}. The density loss and the point number of each point cloud are shown at its bottom.

details such as chair arms and legs. Table 6 compares Unpaired and our USSPA on density loss indicating that our predicted point clouds are more uniform.

Autoencoding refiner. As shown in Figure 6, (c) and (d) are the examples of P_{R_0} and P_{R_2} which is the refinement result of the autoencoding refiner on P_{R_0}. The refiner removes the noises from input shown in blue boxes and generates more points on legs as shown in green boxes, which makes the whole point cloud more uniform and accurate with less density loss.

Real scene data. We also test our USSPA on real objects, as shown in Figure 7. We obtain the point clouds of real objects by the 3D scanner and then sample them to 2048 points as the input to our network. Our network can preserve the right arm of the chair and complete the left arm. The predictions are more accurate and uniform as shown in red boxes. This indicates the generalization ability of our USSPA on real objects.

Visualization of latent codes on multi-category data. We utilize t-SNE [21] to visualize the latent codes f_R and f_A of Unpaired and our USSPA on multi-category data. As shown in Figure 8, ten different colors indicate ten categories where dark colors indicate real scene data and light colors indicate artificial data. Figure 8 (c) and (d) show the t-SNE results of latent codes on chair category only. Benefiting from our feature discriminator and autoencoder, the latent codes of real scene data are similar to artificial data which can be shown from the distribution of dark and light points. The dark and light points of Unpaired are separated while ours are fixed together for each category. Benefiting from our classifier-guided discriminator, the distributions of

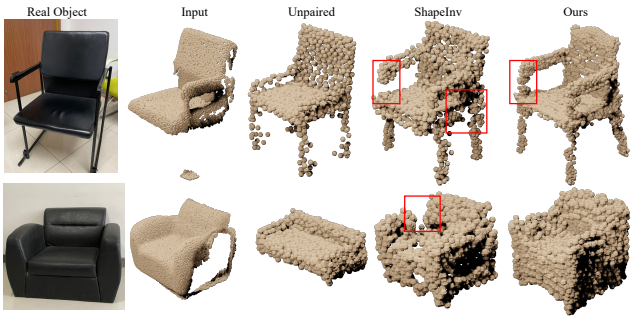

Figure 7. Completion on real objects by Unpaired [32], ShapeInv [36] and our USSPA.

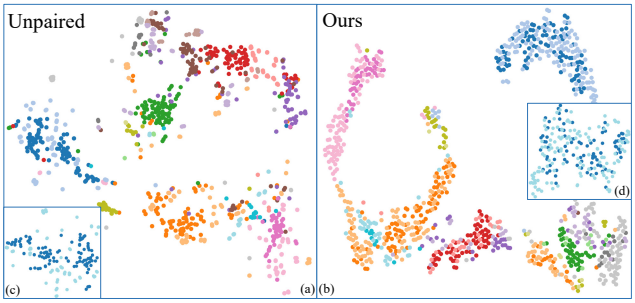

Figure 8. Visualization of t-SNE results of the latent codes by Unpaired [32] and our method. (a) and (b) are the results on multi-category, where different colors denote different categories. Dark and light denote real scene data and artificial data respectively. (c) and (d) are the results on the chair category only.

our latent codes of different categories are separated and clean. For example, as Figure 8 (b) shows, the distances between pink, orange and blue clusters are bigger and the clusters are tighter.

5. Conclusion, Limitation and Future Work

We have presented USSPA, an end-to-end unsupervised network for completion of real scene point cloud objects. Benefiting from our carefully designed symmetry learning module and refinement autoencoder, our prediction preserves the symmetrical shape of input with more accurate and uniform points. We also contribute a real scene dataset for accurate evaluation, and extensive experiments and comparisons show the superiority and generalization of our method on different categories of different datasets and real objects, which achieves state-of-the-art performance on unsupervised completion of real scene objects. For those categories without artificial data as assistance, previous works and our method are hard to apply in this situation. This is our limitation. Unsupervised point cloud completion without the assistance of artificial data is our future work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant numbers 62032011, 61972194.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5

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

[3] Kaichun Mo Leonidas J. Guibas Charles R. Qi, Hao Su. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. 4

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 1, 2, 5

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5, 6

[7] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1359–1368, 2019. 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2, 3

[9] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1

[10] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh. *ACM Transactions on Graphics (TOG)*, 39:126:1 – 126:12, 2020. 1

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

[12] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018. 1

[13] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017. 1

[14] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *2017 Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017. 6, 7

[15] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016. 2

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

[17] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1716–1725, 2022. 1, 2

[18] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 6

[19] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, 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. 1, 2, 5, 6, 7

[20] 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

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 8

[22] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 1, 2

[23] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Learning local displacements for point cloud completion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1558–1567, 2022. 1, 2, 5, 6, 7

[24] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13075–13084, 2021. 1, 3, 5, 6, 7

[25] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 1939–1948, 2020. 2

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

[27] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *The European Conference on Computer Vision (ECCV)*, August 2020. 3

[28] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. 11 2021. 2

[29] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–17, 2022. 3

[30] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[31] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[32] Niloy J. Mitra Xuelin Chen, Baoquan Chen. Unpaired point cloud completion on real scans using adversarial training. In *2020 International Conference on Learning Representations 2020 (ICLR)*, 2020. 1, 3, 5, 6, 7, 8

[33] 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. 1

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

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

[36] Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through gan inversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1768–1777, 2021. 1, 3, 5, 6, 7, 8

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