Multi-Stage Point Completion Network with Critical Set Supervision

Wenxiao Zhang[a], Chengjiang Long[b], Qingan Yan[c], Alix L.H.[d], Chunxia Xiao[a,*]

[a]*School of Computer Science, Wuhan University*
[b]*Kitware Inc., Clifton Park, NY, USA*
[c]*JD.com American Technologies Corporation, CA*
[d]*Xiaomi*

Abstract

Point cloud based shape completion has great significant application values and refers to reconstructing a complete point cloud from a partial input. In this paper, we propose a multi-stage point completion network (MSPCN) with critical set supervision. In our network, a cascade of upsampling units is used to progressively recover the high-resolution results with several stages. Different from the existing works that generate the output point cloud structure supervised by the complete ground truth, we leverage the critical set at each stage for supervision and generate a more informative and useful intermediate outputs for the next stage. We propose a strategy by combining max-pooling selected points and volume-downsampling points to determine critical sets (MVCS) for supervision, which concerns both a critical features and the shape of the model. We conduct extensive experiments on the ShapeNet dataset and the experimental results clearly demonstrate that our proposed MSPCN with critical set supervision outperforms the state-of-the-art completion methods.

Keywords: Shape completion, Point cloud, Deep learning

1. Introduction

An increasingly large volume of 3D data is becoming largely available due to the rapid growth of the 3D scan technology with low-cost sensors like depth camera or LIDAR. However, the acquired scan data is often incomplete due to occlusion and sensor resolution. It is desired to recover a complete shape even from a partial input, which refers to the task of shape completion and has significant values in multiple fields like 3D reconstruction Dai et al. (2017); Liao et al. (2019); Fu et al. (2018); Yan et al. (2017, 2016), robotics Varley et al. (2017), scene understanding Dai et al. (2018) and autonomous driving Yang et al. (2019).

Most existing deep learning methods for shape completion just discretize the 3D data into voxel such as occupied grids or Truncated Signed Distance Function (TSDF) volume where convolution operations can be applied directly. However, the output of these methods is always in low-resolution due to the memory cost of volumetric representation and discards some object details. As a raw representation of 3D objects, point cloud is able to overcome the shortcoming of volumetric representation. Recent, Yuan *et al.* Yuan et al. (2018) proposed the first point completion network using an encoder-decoder network in a coarse-to-fine fashion. Taking point cloud as input, this two-stage network generates a coarse output at the 1st stage and then produces the final result based on the coarse output at the 2nd stage. This motivates us to further explore the idea of the multi-stage refinement in point completion networks, as illustrated in Figure 1.

In this paper, we propose a multi-stage point completion network (MSPCN), as shown in Figure 2. We argue that it is intuitive to progressively recover the complete object shape with multiple stages, where the network firstly generates a low-resolution result and then infers a higher-resolution shape based on the lower-resolution result at the previous stage.

[*] Corresponding author

Email addresses: wenxxiao.zhang@gmail.com (Wenxiao Zhang), cjfykx@gmail.com (Chengjiang Long), qingan.yan@jd.com (Qingan Yan), zhouliheng@xiaomi.com (Alix L.H.), cxxiao@whu.edu.cn (Chunxia Xiao)

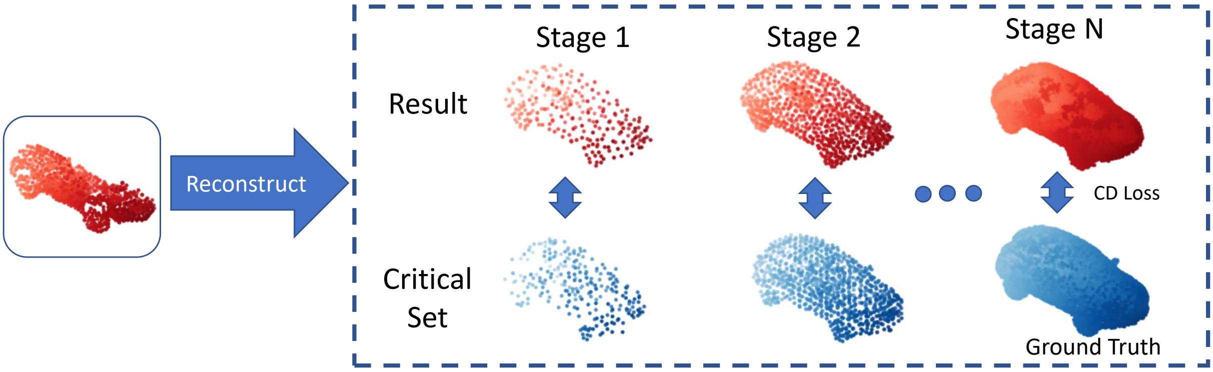

Figure 1: Given an incomplete point cloud (left), our goal is to recover a complete point cloud (right) with a multi-stage network via upsampling from lower-resolution to higher-resolution gradually. Instead of using the complete ground truth point cloud to train the overall network, we introduce a critical set for each stage to conduct intermediate supervision and predict more useful and informative points for the next stage.

It is not reasonable to use the high-resolution complete ground truth at every stage because it will burden the network to figure out the correlation between the the low and high-resolution points, which causes more computation when computing the loss. Moreover, directly using a subsampled point set of the high-resolution ground truth might miss some critical features in the original model. Therefore, we adopt intermediate supervision and introduce *critical set supervision* to determine the corresponding ground truth, instead of all using the complete ground-truth point cloud, for supervision at each intermediate stage. The *critical set* involves the points which play key roles in representing the model shape. This treatment ensures the recovered points at the current stage are more useful and critical for further recovery at the next stage.

To determine a useful critical set including points capturing critical feature and also preserving the shape surface, we propose a strategy of combining max-pooling selected points and volume-downsampling points to determine critical sets for intermediate supervision. We notice that it is very critical to recognize the latent shape of the partial input for the completion task, so we use a self-supervised point cloud auto-encoder network to find the points which are important in reconstructing themselves. A set of points are chosen if their feature values are selected by the max-pooling operation when they are aggregated to the global feature in the auto-encoder network. Points in this set can well represent the most critical features in reconstruction task. Another set of points are the uniformly downsampled points extracted from the complete ground-truth so that they can well represent the shape surface of the model. Then these two sets are mixed with nearest neighbour matching to determine the final critical set. Note that we determine multi-resolution critical sets for intermediate supervision at all stages. Extensive experimental results show this critical set supervision can boost the performance for the point completion task.

To sum up, our main contributions are three-fold: (1) we propose a general multi-stage network (MSPCN) for progressive point cloud completion. It generates a cascade of multi-resolution point clouds as intermediate results, which are used to conduct further completion at the next stage; (2) we propose a combining strategy to determine critical sets for supervision, which explores the critical points extracted in an unsupervised manner; and (3) extensive experiments clearly demonstrate that combined with critical points supervision strategy, our multi-stage network outperforms state-of-the-art 3D point cloud completion methods.

2. Related Work

In this section, we first introduce some recent learning methods on point clouds as our method focus on point clouds object completion, then we briefly review some existing related shape completion methods, which can be divided into two categories, *i.e.*, *non-learning based methods* and *learning based methods*. Finally, we review some works that mention the critical set in point cloud.

Deep learning on point clouds. Qi *et al.* Qi et al. (2017a) first introduced a deep learning network PointNet which uses symmetric function to directly process point cloud. PointNet++ Qi et al. (2017b) captures point cloud local

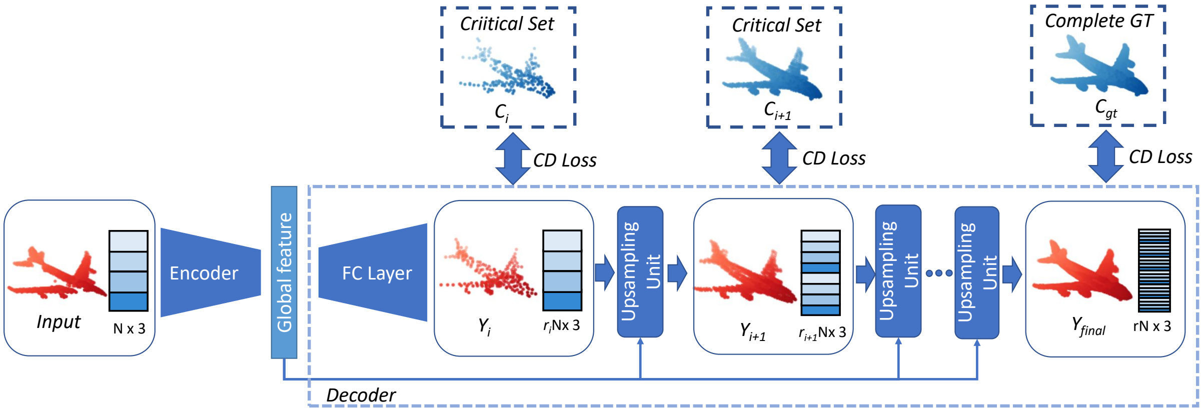

Figure 2: The overall network architecture of our proposed MSPCN with critical set supervision. The network takes a partial input with N points and uses the PointNet based encoder to extract the global feature. Fully connected layers are used to generate a low-resolution result. Then we use a cascade of upsampling units to progressively recover the high-resolution results with several stages. At each stage, we leverage the critical set for supervision to generate an intermediate outputs for the next stage.

structure from neighborhoods at multiple scales. FoldingNet Yang et al. (2018) uses a novel folding-based decoder which deforms a canonical 2D grid onto the underlying shape surface. There are also a series of network architectures Achlioptas et al. (2017); Su et al. (2018); Hua et al. (2018); Lin et al. (2018); Li et al. (2018); Navaneet et al. (2019); Zhao et al. (2019); Li et al. (2019); Zhang and Xiao (2019) on point clouds are proposed in succession for point cloud processing. It is worth mentioning that our proposed MSPCN leverages several recent advances in deep neural networks that directly process point clouds.

Non-learning based shape completion. Shape completion has long been a problem on interest in the graphics and vision. Some classic descriptors have been developed in the early years, such as Nealen et al. (2006); Sorkine and Cohen-Or (2004); Kazhdan and Hoppe (2013), which leverages geometric cues to fill the missing parts in the surface. These methods are usually limited to fill only small holes. Another way to complete the shape is to find the symmetric structure as priors to achieve the completion Thrun and Wegbreit (2005); Pauly et al. (2008); Mitra et al. (2006). However, these methods can only work when the missing part can be inferred from the existing partial model. Some researchers proposed data-driven methods Li et al. (2015); Shi et al. (2016); Kim et al. (2012) which usually retrieve the most likely model based on the partial input from a large 3D shape database. Though convincing results can be obtained, these methods are time consuming in matching process according to the database size.

Learning based shape completion. With the breakthroughs of some learning based 2D vision tasks over the past few years, more and more researchers tend to solve 3D vision tasks using learning methods. Learning based methods on shape completion usually use deep neural network with an encoder-decoder architecture to directly map the partial input to a complete shape. Most pioneering works Wu et al. (2015); Li et al. (2016); Dai et al. (2017); Han et al. (2017); Yang et al. (2017) rely on volumetric representations where convolution operations can be directly applied. Volumetric representations lead to large computation and memory costs, thus most works operate on low dimension voxel grids which causes details missing. To avoid these limitations, Yuan *et al.* proposed PCN Yuan et al. (2018) which directly generates complete shape with partial point cloud as input. PCN recovers the complete point cloud in 2-stage which first generates a coarse result with low resolution and then recovers the final output using the coarse one. Lyne *et al.* propose TopNet Tchapmi et al. (2019) which propose a hierarchical rooted tree structure as decoder to generate arbitrary grouping of points. Though TopNet uses a tree alike decoder with several levels, it does not generate intermediate results in each level. RL-GAN-Net Sarmad et al. (2019) presents a shape completion framework using the reinforcement learning agent to control the GAN generator.

Critical set in point cloud. There are also few previous work mentioned about critical points in point cloud. Qi *et al.* Qi et al. (2017a) define the set of critical points as the points which contribute to the max pooled feature. Dovrat *et al.* Dovrat et al. (2019) design a sampling network to extract the important points for a certain task. However, these works aim to extract the important points of the network input, which can not directly be applied in our task to find

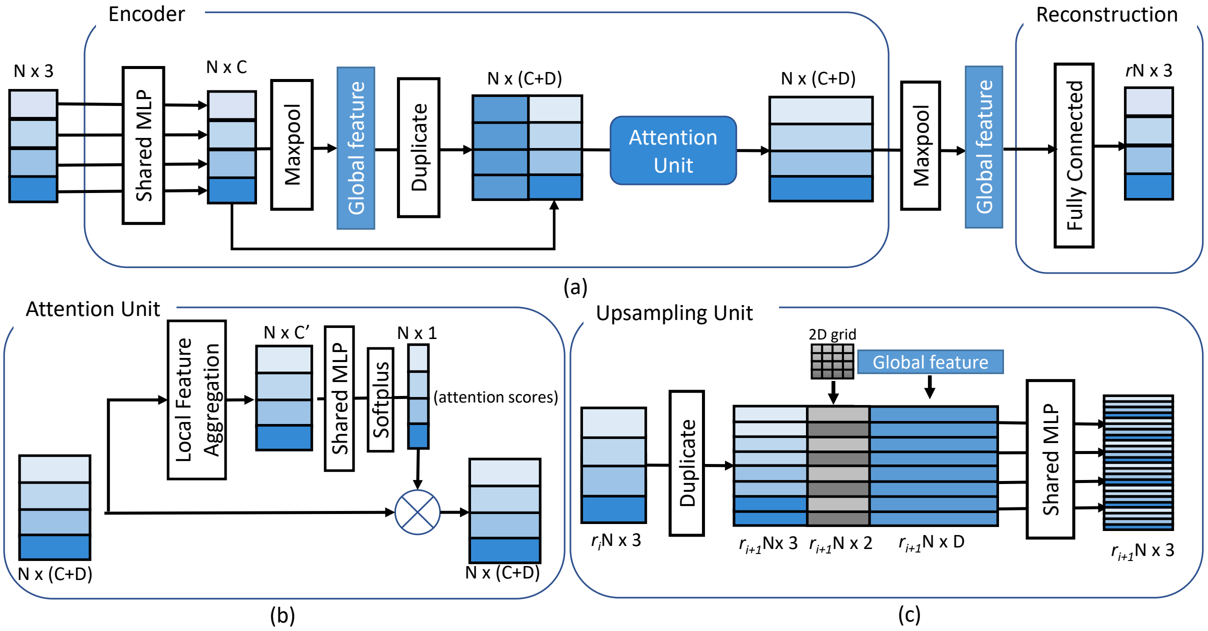

Figure 3: (a) Illustration of the encoder and fully-connected layer architecture. (b) Illustration of the proposed attention unit.(c) Illustration of one upsampling unit.

the critical set in ground truth for supervision.

3. Proposed Approach

In this section, we are going to describe the architecture of our proposed multi-stage network, introduce a novel strategy to determine critical set for intermediate supervision, and discuss the multi-stage training details.

3.1. Overview of the Multi-Stage Network Architecture

Our network is based on PCN Yuan et al. (2018) which is a two-stage network, and we extend it to a general architecture for multi-stage point cloud completion. The network uses an encoder-decoder architecture. The input to our network is a set of d-dimensional points $X_{input} = \{p_i\}_{i \in I} \subset R^d$ and the network generates a point set $Y_{final} = \{p_j\}_{j \in J} \subset R^d$. They both lie on the observed surfaces of an object. The input point set is a partial set of the object obtained from a single view. The output is a uniformly sampled set of the original object. The intermediate output at the i-th stage is defined as Y_i. The whole architecture of our network is shown in Figure 2.

Encoder. We illustrate the encoder architecture in Figure 3(a). The encoder comprises two PointNet Qi et al. (2017a) layers to extract the global feature using point-wise multi-layer perceptron with shared weights and max-pooling operation. Instead of directly aggregating point-wise features to the global feature, we add an attention unit to the encoder architecture. The attention unit tries to allocate lower weights for the insignificant point like outliers or noises, and higher weights for the important points like the points which can more effectively represent the shape surface. The details of attention unit are illustrated in Figure 3(b). The attention unit computes scores for each point feature, and then reweigh each point feature before the max-pooling operation. To obtain the scores for each point feature, we first leverage ball query search to find all points that are within a radius to the query point, and then aggregate these local features to the query point. We use the implementation of PointNet++ Qi et al. (2017b) to achieve local feature aggregation. The final scores for each point are obtained by feeding the aggregated features to shared weights multi-layer perceptron with a following softplus activation function.

Decoder. The decoder can be divided into two parts. In the first part, we use fully connected layers to transform the global feature size from $1 \times (C + D)$ to $1 \times (rN \times 3)$, and then resize it to $rN \times 3$ to reconstruct a low-resolution point clouds which is shown in Figure 3(a). In the second part, we use a cascade of upsampling units to progressively recover the high-resolution result.

Upsampling Unit. The motivation of upsampling unit is to recover a high-resolution result from a low-resolution one. This can be seen as an upsampling process with shape correction. PU-Net Yu et al. (2018) was first proposed for point cloud upsampling task, which uses multi-branch convolution unit to extract features and get high resolution results. However, if we directly use the PU-Net, the generated points are easy to be clustered around the original points positions. In PU-Net, it uses a repulsion loss to let the generated points to be apart from each other. Instead of training the network to force the generated point to be scattered in the space, we explicitly offer the network the information about the position variation. We concatenate the original coordinates with a 2D grid to transform them to different locations which is proposed in FoldingNet Yang et al. (2018). As shown in Figure 3(c), the output at the $i+1$-th stage output can be expressed as:

$$Y_{i+1} = \mathbf{U}(Y_i, C_{grid}, F_{global}) \tag{1}$$

where $\mathbf{U}$ represent the non-linear transformation fitted by upsampling unit, C_{grids} represent the 2D-grid coordinates, F_{global} represent the global feature. We also test the PU-Net as upsampling unit in our experiments part.

Note that since we try to minimize the loss at every stage, the upsampling unit has ability to gradually correct shape error.

Loss Function. To measure the differences between two point clouds (S_1, S_2), Chamfer Distance (CD) and Earth Mover's Distance (EMD) are commonly used in recent researches which are introduced by Fan et al. (2017). We choose Chamfer Distance as our loss function:

$$\mathcal{L}_{CD}(S_1, S_2) = \frac{1}{|S_1|}\sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{|S_2|}\sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2 \tag{2}$$

Since we test several critical set supervision strategies in our experiments, where the number of critical points can not always be controlled to be the same as the network output size at each stage, we choose $\mathcal{L}_{CD}$ where the two sets do not need to be the same size and which is faster than EMD loss.

3.2. Strategy to Determine Critical Sets for Supervision

As our MSPCN uses a multi-stage architecture, it generates a cascade of multi-resolution point clouds as intermediate results. Instead of using the high-resolution complete ground truth at all stages, we propose a critical set supervision in which critical point sets of different resolution are used for supervision at different stage. We believe there exists a point set which can well represent the critical features of the original model and preserve the shape surface for further recovering at the next stage. The quality of critical set directly determines the completion performance. Therefore, it is desirable for us to determine a more informative and representative critical set at each stage.

We propose a strategy, combining *Max-pooling Selected Points* and *Voxel-downsampling Points* to determine *Critical Sets*, denoted as MVCS, for supervision on each upsampling at all stages. Max-pooling operation provides guidance to extract points that capture critical features, and volume-downsampling is to sample a certain number of points that can represent the shape surface roughly on the whole. Therefore, our strategy of combining these two aspects can ensure us to get an informative and representative critical set in which there are points with critical features and points covering the shape surface well. We now introduce how we extract these two point sets.

Max-pooling Selected Points. As is defined in PointNet, max-pooling operation is usually used to aggregate the $N \times D$ point-wise features to the final $1 \times D$ global feature vector. The critical set is defined as those points whose feature values are ”selected” by the max-pooling operation to be included into the final global feature vector.

Inspired by this work, we first extract a point set which can represent the critical features of the model. The main idea is that we consider the key to the completion task is to recognize the latent shape of the model, so we use a self-supervised point cloud auto-encoder network to find the critical points which are important in reconstructing themselves. The auto-encoder network comprises a similar encoder and fully-connected layer to our proposed network, but the input and output are both the ground truth complete point cloud as shown in Figure 4, where we extract the critical points which are selected by the final max-pooling operation.

The number of critical points selected by max-pooling is usually far less than the global feature size, because each point can be selected at most once and only a few points can be selected by max-pooling. We observe that the

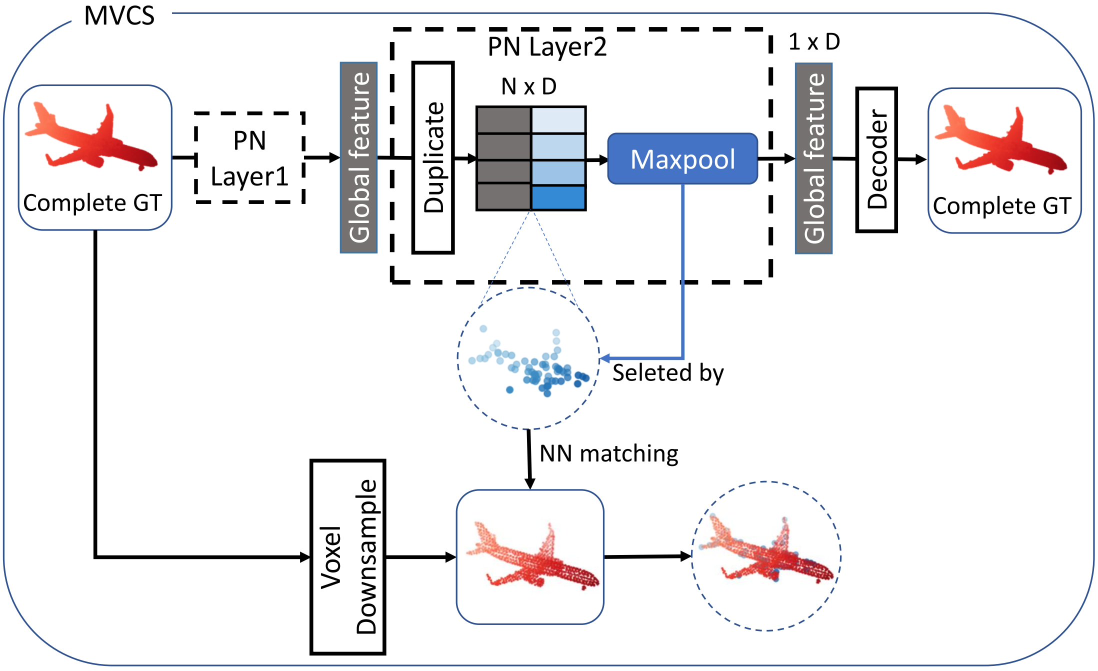

Figure 4: Illustration of points selected by max-pooling operation and the combining process.

average critical set size is about 161 out of 16384 input points where the global feature size is 1024 in our experiments. Obviously, this size is too small to represent the coarse shape surface for supervision at every stage, even though it can collect some critical features of the complete point cloud.

Combining with Voxel-downsampling Points. Voxel-downsampling extract points by applying a uniformly sampling using voxel grid filter Rusu and Cousins (2011). Therefore, the volume-downsampling can well preserve the shape surface of the complete ground-truth point cloud. So we consider combining the max-pooling selected points with the voxel-downsampling points. Denote $\mathbf{S}_i^{mp}$ to be the set of points obtained according to the max-pooling result, and $\mathbf{S}_i^{vd}$ to be the set of points through volume-downsampling at the i-th stage. We mix these two point sets using nearest neighbour matching. For each point $\mathbf{x} \in \mathbf{S}_i^{mp}$, we first find the closest Euclidean corresponding point $\mathbf{y}^* \in \mathbf{S}_i^{vd}$, and then replace the original $\mathbf{y}^*$ with $\mathbf{x}$ to get a set of matching points. *i.e.*,

$$\mathbf{y}^* = \mathrm{NNM}(\mathbf{x}, \mathbf{S}_i^{vd}) = \arg\min_{\mathbf{y}\in\mathbf{S}_i^{vd}} \|\mathbf{x} - \mathbf{y}\|_2, \tag{3}$$

and get a set of points that will be replaced out,

$$\mathbf{S}_i^{nnm} = \{\mathrm{NNM}(\mathbf{x}, \mathbf{S}_i^{vd}) | \mathbf{x} \in \mathbf{S}_i^{mp}\}, \tag{4}$$

where $\mathrm{NNM}(\cdot)$ denotes the nearest neighbour matching operation. Therefore, there is an mapping relationship between $\mathbf{S}_i^{mp}$ and $\mathbf{S}_i^{nnm}$. Note that since it may find more than one corresponding x in S_i^{mp} for the same $y*$, we replace the $y*$ with the closest corresponding x. It means only one x will be retained if some points in S_i^{mp} are too close.

Considering that $\mathbf{S}_i^{vd}$ can be partitioned into $\mathbf{S}_i^{nnm}$ and $\mathbf{S}_i^{vd} - \mathbf{S}_i^{nnm}$, we can get the final critical set at the i-th stage by replacing $\mathbf{S}_i^{nnm}$ with $\mathbf{S}_i^{mp}$, *i.e.*,

$$C_i = (\mathbf{S}_i^{vd} - \mathbf{S}_i^{nnm}) \cup \mathbf{S}_i^{mp}. \tag{5}$$

The obtained C_i can well represent the critical features and approximate the shape surface.

Besides, we consider that recognizing the correct category of the model may also be useful in completion task, thus we also test the performance of using the critical points in classification task in ablation studies. The classification network is similar to our network encoder where we use the final global feature to obtain the per-class score.

3.3. *Multi-Stage Training*

With the result Y_i and the critical set C_i at the i stage. The Chamfer Loss at i stage is defined as:

$$\mathcal{L}_i = \mathcal{L}_{CD}(Y_i, C_i) \tag{6}$$

If the network includes m stage, the total loss is:

$$\mathcal{L}_{total} = \sum_{i=1}^{m} \alpha_i \cdot \mathcal{L}_i \tag{7}$$

where $\{\alpha_1 \ldots \alpha_m\}$ are parameters which are manually set during the training process. We first initialize all the parameters to zero before the training. During training procedure, we set $\alpha_i = i$ after $(i-1) \cdot n$ epochs. It can be formulated as follows:

$$\alpha_i = \begin{cases} i & epoch \geq (i-1) \cdot n \\ 0 & epoch < (i-1) \cdot n \end{cases} \tag{8}$$

Namely we train the first i stages for n epochs and then we release the next stage and train the first $i+1$ stages for another n epochs. We just gradually increase the α to force the network to focus more on the new released stage. Note that we train the first stage for $2n$ epochs, since the first output is crucial for the following upsampling.

4. Experiments

In this section, we evaluate our proposed network with critical set supervision strategies quantitatively and qualitatively. We first conduct ablation studies to prove the efficiency of the proposed supervision strategies and then compare our method with state-of-the-art point cloud completion methods on a subset of ShapeNet dataset in terms of the distance metric Chamfer Distance. Note that the Chamfer distance is reported multiplied by 10^4.

The dataset is a subset of the ShapeNet dataset. The complete point cloud contains 16384 points uniformly sampled from the mesh and the partial point clouds with 2048 points are generated by back-projecting 2.5D depth images into 3D. The dataset comprises 8 classes: airplane, cabinet, car, chair, lamp, sofa, table, and vessel. The training set contains 28974 different models. Each model contains a complete point cloud and 3-7 partial point clouds taken from different view. The valid and testing set contains 100 and 1200 point clouds respectively. Note that TopNet Tchapmi et al. (2019) also provides a public completion dataset which is another subset of ShapeNet. However, only the ground truth with 2048 points are available online so that we can not use it to run experiments.

In our experiments, we train our model for 50 epochs with a batch size of 16. The initial learning rate is set to be 0.0007 which is decayed by 0.7 every 50,000 iterations.

4.1. *Supervision Strategies Analysis*

We first evaluate the proposed critical set supervision strategy. We use the network with two-stage as baseline to see the effects of applying different supervision strategies. The two-stage model outputs 1024 points in the first stage and 16384 points as the final output. We run experiments with the following supervision strategies as baselines:

- **GT**: This strategy refers to directly using the ground truth complete point cloud for supervision in all stages.
- **VD**: This strategy uses the point clouds of different resolution uniformly downsampled using voxel grid filter.
- **MCS**: This strategy refers to only using the critical points selected by max-pooling operation to supervise all the intermediate stages.
- **LCS**: Dovrat *et al.* Dovrat et al. (2019) propose a method to learn a subset of the point cloud which can achieve best performance in a certain task. We use this network to learn the useful points in reconstruction task. An overall process is shown in Figure 5. The task net in our experiments refer to the classification and reconstruction task. Unlike the MCS, the critical set size can be controlled by the sampling network.

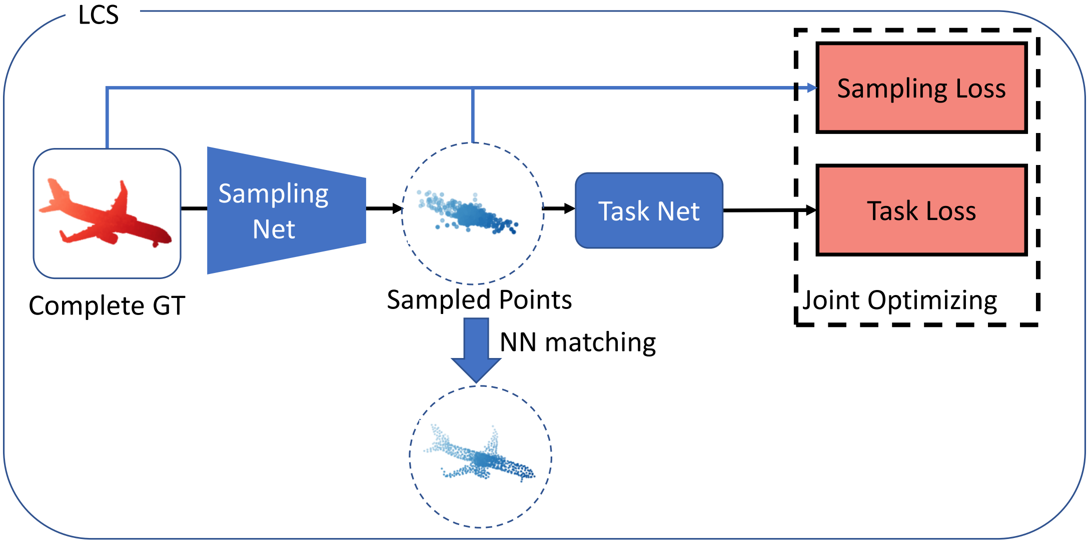

Figure 5: Illustration of the progress of learning to sample the critical set.

Method	airplane	cabinet	car	chair	lamp	sofa	table	vessel	Average
MSPCN w/ GT	5.50	10.57	**8.67**	10.98	11.27	11.69	8.56	9.64	9.61
MSPCN w/ VD	**5.49**	10.64	8.93	10.95	11.02	11.63	8.59	9.45	9.58
MSPCN w/ MCS-AE	6.21	11.23	9.29	11.77	12.77	12.63	9.73	10.13	10.47
MSPCN w/ LCS-AE	5.82	10.77	9.08	11.49	11.46	12.22	9.11	9.77	9.97
MSPCN w/ MVCS-AE	5.55	**10.35**	8.78	**10.94**	**10.95**	**11.51**	**8.51**	**9.42**	**9.50**
MSPCN w/ LCS-CLS	5.79	11.09	9.17	11.57	12.15	12.46	9.39	9.99	10.20
MSPCN w/ MCS-CLS	6.27	11.39	9.41	12.29	12.84	12.38	9.87	10.37	10.60
MSPCN w/ MVCS-CLS	5.59	10.36	8.72	11.03	11.27	11.66	8.61	9.57	9.61

Table 1: Evaluation about the supervision strategy with the metric as Chamfer Distance. The Chamfer distance is reported multiplied by 10^4.

Note that there are also common downsampling methods like farthest point sampling and random sampling. We do not choose these two because the sampled points from these two techniques are random. i.e. FPS algorithm is random and depends on which point is selected first. It means the sampled points may vary in each training iteration when use an online subsampling. While voxel subsampling can ensure the subsampled points are stable.

Since we test the critical set on both self-reconstruction and classification network. We denote these networks respectively. MCS, MVCS and LCS trained on self-reconstruction task based on auto-encoder are denoted as MCS-AE, MVCS-AE and LCS-AE. MCS, MVCS and LCS trained on classification task are denoted as MCS-CLS, MVCS-CLS and LCS-CLS. Note that we train the auto-encoder network for 500 epochs using the completion training set. The final average CD loss on validation set is 6.05×10^4. The classification network is trained for 250 epochs with final average accuracy 97.97%.

From the results in Table 1, we can see the MVCS-AE achieves the best performance and other supervision strategies like VD also improve the results. VD can keep the shape surface to a big extent during the downsampling, but it might discard some points which are critical in representing the model. Combining MCS with VD can exactly complement this defect. We consider this is the reason why MVCS-AE can outperform VD. However, mixing the critical set on classification task with uniform downsampled points (MVCS-CLS) has not improved the results of only using VD. We consider this result attributes to two aspects. On one hand, every class includes a large amount of models with different shape and the aim of completion task is to recover the specific shape instead of the general class. So the classification information may not help in the completion task. On the other hand, the dataset only has 8 classes which is not difficult for the network itself to recognize the correct class.

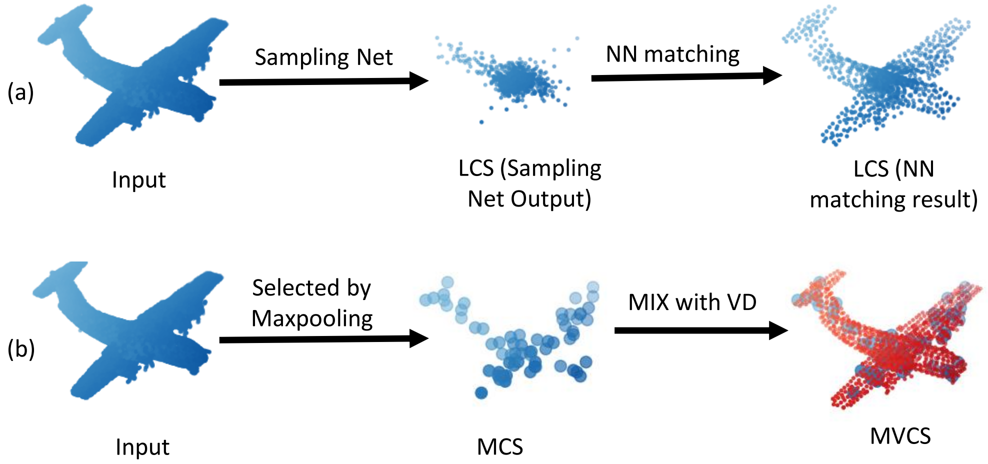

Figure 6: (a) The intermediate result of LCS. (b) The selected points by maxpooling operation and the points mixed with voxel downsampled uniform points.

It's out of our expectation that LCS produces bad results. To find the reason why LCS can not work, we explore some inter-results of LCS in Figure 6(a). Since the points generated by sampling network are not guaranteed to be a subset of the complete ground truth, nearest neighbour matching is applied to match the generated points to the complete ground truth. We can see that the original learned points can not converge to a certain shape. After the matching process, the matched points can approximate the shape better, but there are many redundant points (at the shape center). This is due to the sampling network may can generate points which contribute the max-pooling layer to keep the loss to the minimum, but it fail to keep other generated points to be uniformly spread on the shape surface since other points do not effect the final loss. Compared with LCS, the critical set produced by MVCS shown in Figure 6(b) can better keep both the critical features and the shape surface.

4.2. Multi-Stage Analysis

We test the efficiency of the number of completion steps. Since MVCS-AE achieves the best performance, so we use this supervision strategy in this experiment. In Table 2, we test our proposed network with 2 to 4 stages. The final output point cloud size is 16384 for all models. The 2-stage model is the same to PCN where the output at the first stage has 1024 points. The 3-stage model is with 1024, 4096 points at the 1-2 stages. The 4-stage model is with 256, 1024, 4096 points at 1-3 stages. Note that in the 4-stage network, we use the 256 uniformly sampled points at 1st for supervision and use the MVCS-AE strategy in other stages. From the results we can see the performance is gradually improved with the number of completion stages increasing.

We do not try five or more stages mainly due to GPU memory limitation. The four-stage model can be trained with batch size 16 with about 10 GB GPU memory, while the five-stage model can not be trained with the same setting for memory limitation (we use GTX 2080Ti 11GB).

Method	airplane	cabinet	car	chair	lamp	sofa	table	vessel	Average
MSPCN w/ MVCS-AE (2 Stages)	5.55	10.35	8.78	10.94	10.95	11.51	8.51	9.42	9.50
MSPCN w/ MVCS-AE (3 Stages)	**5.50**	**10.33**	**8.66**	10.98	10.91	11.48	8.50	9.47	9.48
MSPCN w/ MVCS-AE (4 Stages)	5.59	10.39	8.79	**10.78**	**10.72**	**11.19**	**8.42**	**9.28**	**9.39**

Table 2: Evaluation about the number of the completions steps with the metric as Chamfer Distance. The Chamfer distance is reported multiplied by 10^4.

4.3. Comparison with the State-of-the-art Methods

We qualitatively and quantitatively compare our network with several state-of-the-art point cloud completion methods: FC, FoldingNet Yang et al. (2018), PCN Yuan et al. (2018), and TopNet Tchapmi et al. (2019). For FC, Folding, PCN, we used the pre-trained model and evaluation codes released in the public project of PCN on github. For TopNet, we use their public code and retrain their network of 8 tree levels using the mentioned dataset. Table 3 shows the quantitative comparison results. Our proposed network achieves the lowest values for the evaluation metrics with the critical supervision strategy in most categories. However, our method seems do not improve the performance on car and airplane categories. The reason we find is that most of the models in car and airplane category have similar skeletons. The details of generated model will not effect the CD loss much as long as the generated model has a correct skeleton. However, models in other categories are diverse without a stable skeleton just like various types of lamp in lamp category.

Besides quantitative results, some qualitative examples from the testing set are shown in Figure 7. We can see the results generated by our network are more stable with less distortion, while other methods seem to suffer from more noises.

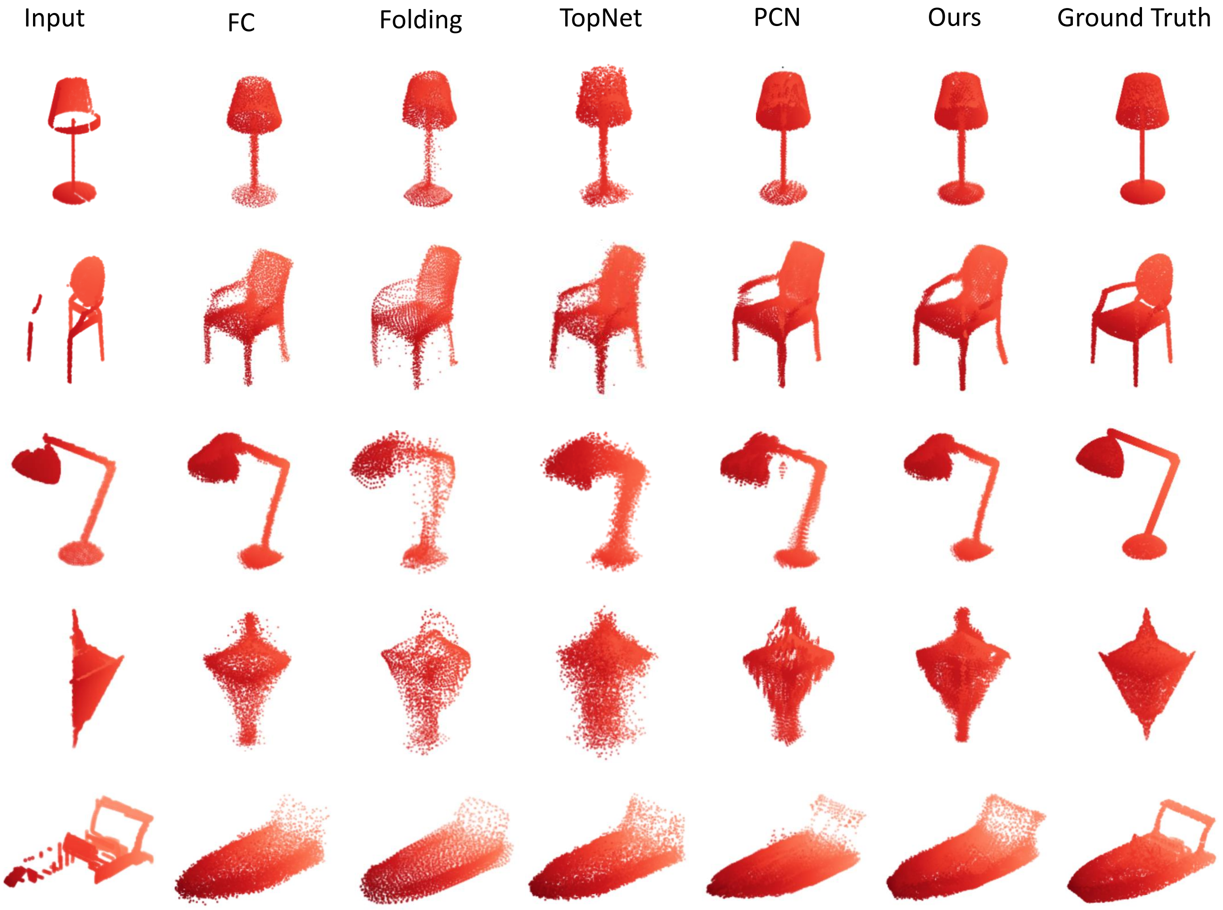

Figure 7: Qualitative comparison with state-of-the-art methods.

4.4. Ablation studies

Attention Unit Analysis To evaluate the effectiveness of the attention unit, we remove the attention unit from the encoder and keep other components unchanged for comparison. Evaluation results are shown in Table 4. We can

Method	airplane	cabinet	car	chair	lamp	sofa	table	vessel	Average
FC	5.70	11.02	8.78	10.97	11.13	11.76	9.32	9.72	9.80
Folding	5.98	10.98	9.36	11.16	11.87	11.42	9.28	9.61	9.96
PCN	**5.51**	10.62	**8.67**	11.00	11.34	11.68	8.59	9.67	9.64
TopNet	5.85	10.78	8.84	10.80	11.15	11.41	8.79	9.41	9.63
MSPCN w/ MVCS	5.59	**10.39**	8.79	**10.78**	**10.72**	**11.19**	**8.42**	**9.28**	**9.39**

Table 3: Quantitative comparison with state-of-the-art methods with the metric as Chamfer Distance. The Chamfer distance is reported multiplied by 10^4.

observe that removing the attention unit will reduce the overall performance, meaning that the proposed attention unit contributes.

Method	With attention unit	Without attention unit
MSPCN w/ MVCS-AE (2 Stages)	**9.50**	9.61
MSPCN w/ MVCS-AE (3 Stages)	**9.48**	9.50
MSPCN w/ MVCS-AE (4 Stages)	**9.39**	9.45

Table 4: Evaluate of the attention unit with the metric as Chamfer Distance. The Chamfer distance is reported multiplied by 10^4.

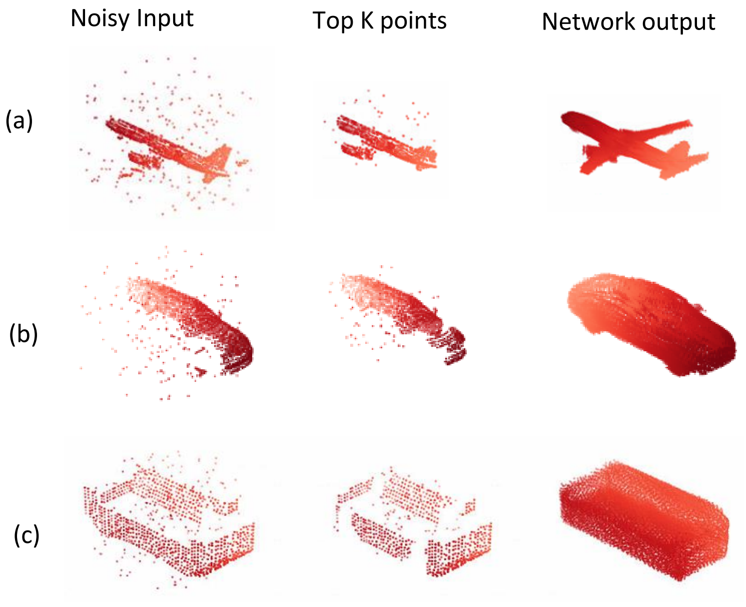

Figure 8: Visualization of top K weights points.

As we test our methods on synthetic data, to verify the statement “the attention unit tries to allocate lower weights for the insignificant point like outliers or noises”, we random add some noises to the data. Given the input with N points, we replace the $(1 - K)$ points with noises. We visualize the top K points which get the highest weights, with results shown in Figure 8. In our experiments, we set $N = 2048$ and K=$0.75N$. It can be observed that the points with high weights filter most of the noises and can preserve the input surface.

Upsampling Unit Analysis We test different upsamping units. Besides the folding net unit, we also test the PU-Net Yu et al. (2018), which is a network proposed for point cloud upsampling. We use the implements of the public code of PU-Net. The main difference between the folding net unit and PU-Net unit is that PU-Net needs to use a repulsion loss to let the generated points to be apart from each other. We select some completion results with similar

CD loss in Figure 9. We observe that although PU-Net uses a repulsion loss, the generated points are still tangled to the original points to some extent. However, with the 2D-grid concatenated to the original coordinates, the generated points of FoldingNet are uniformly distributed in the latent surface.

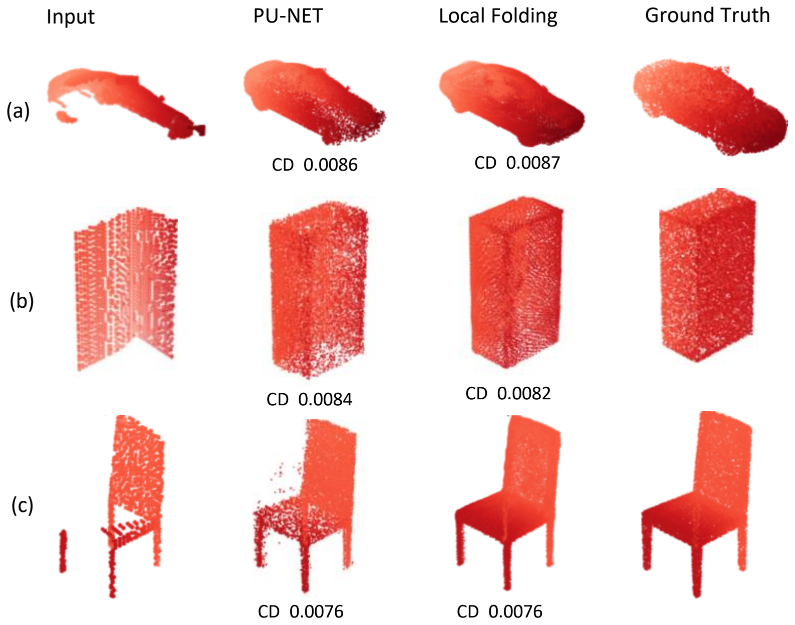

Figure 9: Qualitative comparison of the results produced by using PU-NET and local folding unit.

We also consider to try other upsampling method like MPU Yifan et al. (2019), but it uses multi-level upsampling strategy which also involves feature interpolation between different level, which is not light-weighted enough to be integrated to our network. The proposed training process in MPU is also complex.

5. Conclusion

In this paper, we propose a multi-stage point completion network (MSPCN) with a novel critical set supervision strategy to boost the completion performance. We use the points which are more important in the reconstruction task and then combine them with the uniformly downsampled point set to keep both the critical features and the model shape. The supervision strategy is applied in a multi-resolution manner. Experiments show our proposed multi-stage point completion network performs better than state-of-the-art point cloud completion methods quantitatively and qualitatively. We plan to extent our methods on real-scanned data as a future work.

6. Acknowledgements

We thank all the anonymous reviewers for their insightful comments and feedback. This work was partly supported by Key Technological Innovation Projects of Hubei Province (2018AAA062), Wuhan Science and Technology Plan Project (No. 2017010201010109), the National Key Research and Development Program of China (2017YFB1002600), the NSFC (No. 61672390, 61972298). The corresponding author is Chunxia Xiao.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2017. Representation learning and adversarial generation of 3d point clouds. arXiv preprint arXiv:1707.02392 2, 4.

Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J., Nießner, M., 2018. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans, in: CVPR, pp. 4578–4587.

Dai, A., Ruizhongtai Qi, C., Nießner, M., 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis, in: CVPR, pp. 5868–5877.

Dovrat, O., Lang, I., Avidan, S., 2019. Learning to sample, in: CVPR.

Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3d object reconstruction from a single image, in: CVPR, pp. 605–613.

Fu, Y., Yan, Q., Yang, L., Liao, J., Xiao, C., 2018. Texture mapping for 3d reconstruction with rgb-d sensor, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 4645–4653.

Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y., 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference, in: ICCV, pp. 85–93.

Hua, B.S., Tran, M.K., Yeung, S.K., 2018. Pointwise convolutional neural networks, in: CVPR, pp. 984–993.

Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. ToG 32, 29.

Kim, Y.M., Mitra, N.J., Yan, D.M., Guibas, L., 2012. Acquiring 3d indoor environments with variability and repetition. TOG 31, 138.

Li, D., Shao, T., Wu, H., Zhou, K., 2016. Shape completion from a single rgbd image. TVCG 23, 1809–1822.

Li, R., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A., 2019. Pu-gan: A point cloud upsampling adversarial network, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 7203–7212.

Li, R., Wang, S., Zhu, F., Huang, J., 2018. Adaptive graph convolutional neural networks, in: AAAI.

Li, Y., Dai, A., Guibas, L., Nießner, M., 2015. Database-assisted object retrieval for real-time 3d reconstruction, in: CGF, Wiley Online Library. pp. 435–446.

Liao, J., Fu, Y., Yan, Q., Xiao, C., 2019. Pyramid multi-view stereo with local consistency. Computer Graphics Forum 38, 335–346.

Lin, C.H., Kong, C., Lucey, S., 2018. Learning efficient point cloud generation for dense 3d object reconstruction, in: AAAI.

Mitra, N.J., Guibas, L.J., Pauly, M., 2006. Partial and approximate symmetry detection for 3d geometry, in: TOG, ACM. pp. 560–568.

Navaneet, K., Mandikal, P., Agarwal, M., Babu, R.V., 2019. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision, in: AAAI, pp. 8819–8826.

Nealen, A., Igarashi, T., Sorkine, O., Alexa, M., 2006. Laplacian mesh optimization, in: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, ACM. pp. 381–389.

Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H., Guibas, L.J., 2008. Discovering structural regularity in 3d geometry, in: TOG, ACM. p. 43.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 1, 4.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: NeurIPS, pp. 5099–5108.

Rusu, R.B., Cousins, S., 2011. 3d is here: Point cloud library (pcl), in: ICRA, IEEE. pp. 1–4.

Sarmad, M., Lee, H.J., Kim, Y.M., 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion, in: CVPR, pp. 5898–5907.

Shi, Y., Long, P., Xu, K., Huang, H., Xiong, Y., 2016. Data-driven contextual modeling for 3d scene understanding. Computers & Graphics 55, 55–67.

Sorkine, O., Cohen-Or, D., 2004. Least-squares meshes, in: Proceedings Shape Modeling Applications,, IEEE. pp. 191–199.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J., 2018. SPLATNet: Sparse lattice networks for point cloud processing, in: CVPR, pp. 2530–2539.

Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S., 2019. Topnet: Structural point cloud decoder, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 383–392.

Thrun, S., Wegbreit, B., 2005. Shape from symmetry, in: ICCV, IEEE. pp. 1824–1831.

Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P., 2017. Shape completion enabled robotic grasping, in: IROS, IEEE. pp. 2442–2447.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes, in: CVPR, pp. 1912–1920.

Yan, Q., Yang, L., Liang, C., Liu, H., Hu, R., Xiao, C., 2016. Geometrically based linear iterative clustering for quantitative feature correspondence, in: Computer Graphics Forum, Wiley Online Library. pp. 1–10.

Yan, Q., Yang, L., Zhang, L., Xiao, C., 2017. Distinguishing the indistinguishable: Exploring structural ambiguities via geodesic context, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3836–3844.

Yang, B., Wen, H., Wang, S., Clark, R., Markham, A., Trigoni, N., 2017. 3d object reconstruction from a single depth view with adversarial learning, in: ICCV, pp. 679–688.

Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B., 2019. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios, in: CVPR.

Yang, Y., Feng, C., Shen, Y., Tian, D., 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation, in: CVPR.

Yifan, W., Wu, S., Huang, H., Cohen-Or, D., Sorkine-Hornung, O., 2019. Patch-based progressive 3d point set upsampling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5958–5967.

Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A., 2018. Pu-net: Point cloud upsampling network, in: CVPR, pp. 2790–2799.

Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M., 2018. Pcn: Point completion network, in: 3DV, IEEE. pp. 728–737.

Zhang, W., Xiao, C., 2019. Pcan: 3d attention map learning using contextual information for point cloud based retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12436–12445.

Zhao, H., Jiang, L., Fu, C.W., Jia, J., 2019. Pointweb: Enhancing local neighborhood features for point cloud processing, in: CVPR, pp. 5565–5573.