IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 28, 2019, accepted September 16, 2019, date of publication September 23, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943235

Attention-Based Dense Point Cloud Reconstruction From a Single Image

QIANG LU[1,2,3], **MINGJIE XIAO**[2], **YIYANG LU**[2], **XIAOHUI YUAN**[4], **(Senior Member, IEEE), AND YE YU**[1,2,3]

[1]Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, Hefei 230009, China
[2]School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[3]Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei 230009, China
[4]Department of Science and Engineering, University of North Texas, Denton, TX 76203, USA

Corresponding author: Qiang Lu (luqiang@hfut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972130 and Grant 61906061, in part by the Research and Development Plan of Anhui Province under Grant 1704d0802177, in part by the Provincial Key Research and Development Program of Anhui Province under Grant 201904d07020010, in part by the Natural Science Foundation of Anhui Province of China under Grant 1708085MF158, in part by the Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture of China, under Grant AEC2018003, and in part by the Key Project of Transformation and Industrialization of Scientific and Technological Achievements of Intelligent Manufacturing Technology Research Institute of Hefei University of Technology under Grant IMICZ2017010.

ABSTRACT Three-dimensional Reconstruction has drawn much attention in computer vision. Generating a dense point cloud from a single image is a more challenging task. However, generating dense point clouds directly costs expensively in calculation and memory and may cause the network hard to train. In this work, we propose a two-stage training dense point cloud generation network. We first train our attention-based sparse point cloud generation network to generate a sparse point cloud from a single image. Then we train our dense point cloud generation network to densify the generated sparse point cloud. After combining the two stages and finetuning, we obtain an end-to-end network that generates a dense point cloud from a single image. Through evaluation of both synthetic and real-world datasets, we demonstrate that our approach outperforms state of the art works in dense point cloud generation. Our source code is available at https://github.com/VIM-Lab/AttentionDPCR.

INDEX TERMS 3D reconstruction, point-cloud, attention mechanism, two-stage training, single view reconstruction.

I. INTRODUCTION

In the field of computer vision, 3D reconstruction from images is a higher-level task than an image classification task, because computers not only recognize objects but also model objects. Although human vision is flat, after years of learning and cognition, when seeing an object, a human can not only recognize the class and composition of the object but also predict its possible 3D shape. Therefore, excellent machine vision should also be able to have 3D visual perception which makes machine feel the 3D shape from the plane vision. Furthermore, machines can interact with objects when they can model objects, such as robotic arm applications.

With the great success of deep learning in 2D image processing, many researchers tried to duplicate this success into 3D shape processing. Voxel-based 3D reconstruction methods were first proposed because voxels behave like 2D images with regular and ordered structures and distributions. Therefore, simply expanding a 2D convolution into a 3D convolution can be well used in CNN for voxel data. Prior studies [1]–[3] explored the reconstruction methods of 3D voxel grids. For each voxel in 3D grids, the network predicts a probability score of whether or not the voxel is occupied, thereby obtaining a voxel 3D occupancy grid. However, each grid in 3D volumetric representation contains sparse information while a large number of voxels inside are unhelpful in describing the surface feature of 3D shapes. Therefore, 3D volumetric representation is expensive and wasteful in calculation and memory which will grow at the cubic level, especially with increasing resolution. 3D convolution also consumes a lot when extracting features and sampling, which also results in low and limited resolution for most 3D voxel reconstruction tasks.

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

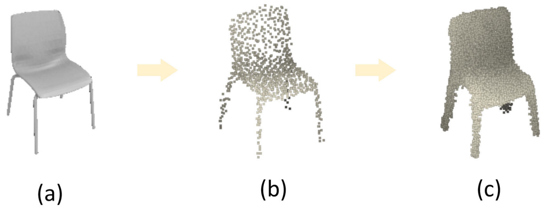


FIGURE 1. Point Clouds generated from a single image. (a) A single RGB image. (b) A sparse point cloud generated from the image. (c) A dense point cloud generated from the sparse point cloud.

Compared with 3D voxels, point clouds are more efficient to represent 3D shapes. Point clouds do not have a problem with the sparsity of information because each of their points effectively describes rich location information. Besides, point clouds are sampled from surfaces of objects so that point clouds capture the details of surfaces of objects while there are no extra internal points which are useless for representing 3D shapes. Fan *et al.* [4] firstly proposed a point set generation network which generated point clouds from single images and proposed an effective metric to measure the distance between two point sets. Feature extraction and convolution operations for point clouds were also introduced by Qi *et al.* [5], [6].

In addition, since 3D reconstruction is based on image processing, the primary task of 3D reconstruction is to extract useful and important features from images. Thus, it is also inseparable from related works in image processing especially image classification which has achieved ideal results through many researchers' works. This lays a solid foundation for 3D reconstruction tasks and the possibility of development. However, most existing 3D reconstruction methods apply vanilla encoders such as VGGNet [7] to extract image features without taking into account that different components of objects in images are not equally important or obvious. Besides, directly generating dense point clouds has some disadvantages. Predicting 16k point coordinates is a large scale regression task which may cause the network more complex and hard to train. The existing dataset is insufficient to train the network. It is also very difficult to apply computationally heavy loss, EMD, to such many points.

In this work, we propose a two-stage training attention-based dense point cloud generation network. Firstly, we extract image features and generate sparse point clouds of 1024 points by introducing attention-based encoder. The image encoder based on the attention mechanism will enhance features of details of objects in images and obtain better features to get higher quality predictions. Then we generate dense point clouds of 16384 points from sparse point clouds through dense point cloud generation network which consists of two dense modules. Finally, we combine these two stages and finetune the network to obtain dense point clouds from single images.

In summary, our contributions in this work are as follows:

- We propose a two-stage training network for generating a dense point cloud, which is densifying a sparse point cloud generated from a single image into a dense point cloud.
- We introduce an encoder based on attention mechanism which makes the network pay more attention to details of shape features in the process of generating sparse point clouds and obtain better reconstruction results.
- We demonstrate that our dense point cloud generation network can directly generate high quality 16x denser point clouds without any intermediate inputs or outputs.
- We evaluate our network on the ShapeNet [8] dataset and the Pix3D [9] dataset and highlight the efficacy of our work in generating dense point clouds on synthetic and real-world datasets, which outperforms the state-of-the-art reconstruction methods.

II. RELATED WORK

A. 3D RECONSTRUCTION

The task of three-dimensional reconstruction from a single image is being studied by more and more researchers in recent years. Due to significantly successful works of 2D CNN in the field of 2D images processing, some researchers attempt to generate 3D voxels using 3D CNN. Wu *et al.* [1] took 2.5D depth maps as input and adopted Gibbs sampling to predict 3D shapes. Choy *et al.* [2] used a 3D recurrent neural network to map multi-view or single-view 2D images to 3D voxels. Gridhar *et al.* [3] learned the embedding of hidden layers by 3D voxel self-encoder to match the corresponding 2D images, and then decoded and generated 3D voxels. Some works [10]–[12] also used octree to organize voxel data to relatively efficiently operate and reconstruct 3D voxels.

Fan *et al.* [4] proposed a network for generating a point set from a single image and an effective metric that measures the distance between two point sets. The results of point set generation network outperformed volumetric approaches [2]. Deformation networks [13], [14] utilized the characteristic of point clouds to be easily deformed. The network learned a deformation matrix to move the control points of matched templates to deform the templates to the final reconstruction results. Mandikal *et al.* [15] proposed a latent matching network which combines image encoding with point cloud encoding for point cloud reconstruction. Zhang *et al.* [16] reconstructed point clouds from single images with complex background by combining nearest point clouds features retrieved from the synthetic dataset. There were also some works [17]–[21] that attempted to reconstruct point clouds by utilizing 2D supervision such as projection, silhouettes and depth maps for 3D reconstruction. Some works [22]–[25] generate 3D meshes from images by deforming primitive meshes, ellipsoid meshes or prepared mesh templates.

B. POINT CLOUD PROCESSING

Point clouds are unordered and distributed in non-regular spaces so they cannot be manipulated by using convolutional neural networks directly. Qi *et al.* [5] proposed PointNet which aligned unordered point cloud and point cloud feature

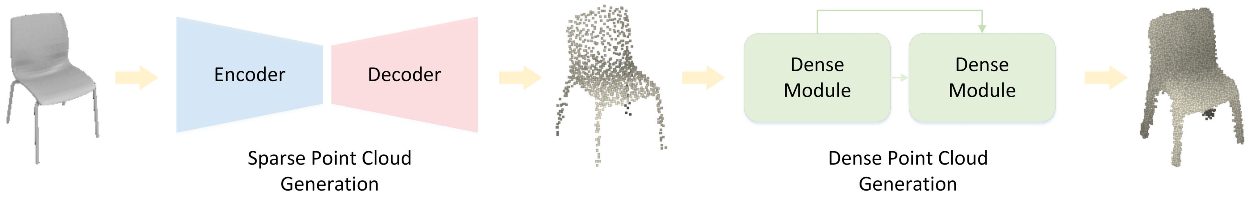


FIGURE 2. Overview of our network structure.

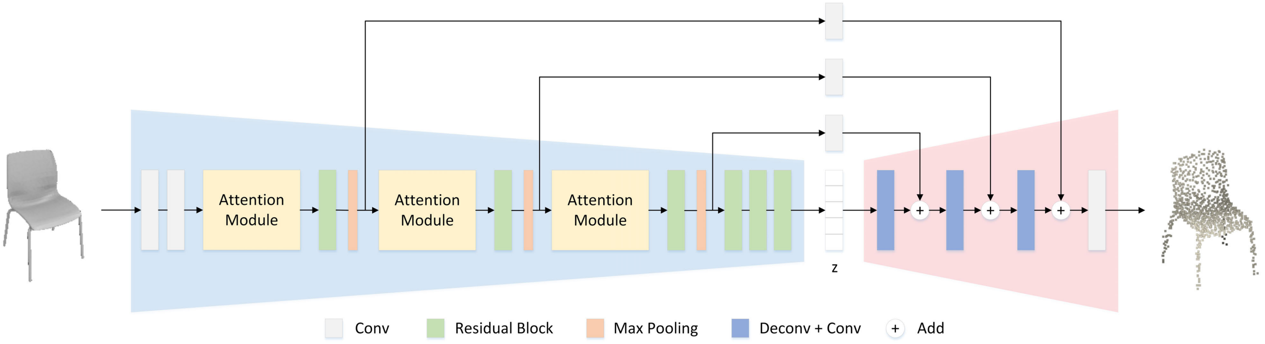


FIGURE 3. Sparse point cloud generation network.

through a simple spatial transformation network and used multi-layer perceptron and global pooling to extract point cloud features. Then Pointnet++ [6] integrated global and local features with a hierarchical feature learning structure. Wang *et al.* [26] proposed edge convolution that extracted local features by finding k nearest neighbor in both point and feature space. Li *et al.* [27] proposed SO-Net that can simulate the spatial distribution of point clouds by constructing self-organizing maps. Wu *et al.* [28] proposed PointConv and corresponding PointDeconv which extended traditional image convolution into point clouds. Yu *et al.* [29] proposed PU-Net to upsample uniform point clouds.

Mandikal and Radhakrishnan [30] proposed a deep pyramidal network for point cloud reconstruction, DensePCR, which is more related to our work. DensePCR hierarchically predicted point clouds of increasing resolution by first predicting a low-resolution point cloud and then increasing its resolution twice, each 4x denser, to a 16x denser high-resolution point cloud. However, there are two key differences between our work and DensePCR. First, DensePCR adopted a commonly used encoder, VGGNet [7], to generate a point cloud from a single image. In contrast, our encoder introduces attention mechanism that allows the network to pay more attention to details of objects in images, extracting better features and thus obtaining better reconstruction results. Second, DensePCR generates a four times denser point cloud through its dense reconstruction network. In order to generate dense point clouds of 16k points, DensePCR needs to train two dense reconstruction networks which first generates 4k points from 1k points and then generates 16k points from 4k points. The former training stage needs extra ground truth point clouds of 4k points to compute loss. Besides, two dense reconstruction networks of DensePCR have no communication, making information transfer more difficult. In contrast, our network can directly predict a 16 times denser point cloud without any extra intermediate input or output.

III. APPROACH

Our goal is to generate dense point clouds from single images. In particular, we first generate a sparse point cloud of 1024 points from a single image and then densify it to a dense point cloud of 16384 points. Our attention-based dense point cloud reconstruction network is shown in Figure 2. A single input RGB image is passed through an attention-based encoder and a decoder consists of a set of deconvs that outputs a sparse point cloud. Then the sparse point cloud is subsequently passed through our dense point cloud generation network to generate a 16x denser point cloud without extra intermediate prediction or input. In this section, we describe two stages of our approach and our training strategy in detail.

A. ATTENTION-BASED SPARSE POINT CLOUD GENERATION

As shown in Figure 3, our training pipeline begins with generating sparse point clouds from single images. Our sparse point cloud generation network encodes an image to feature z and decodes z to a sparse point cloud. Image processing tasks has achieved great success with the help of attention mechanism, while the first stage of 3D reconstruction is extracting features from images. Inspired by [31]–[33], we build an

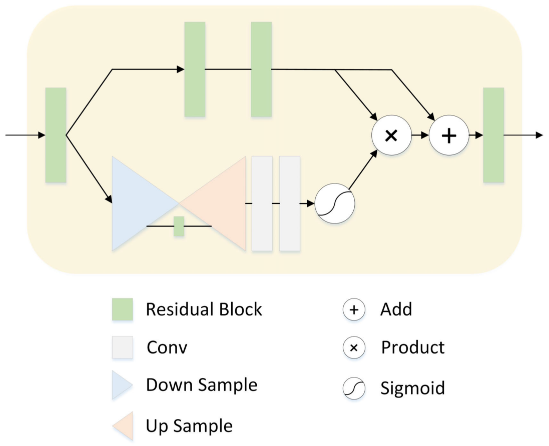


FIGURE 4. Attention module. The top branch is the trunk branch that consists of two residual blocks. The bottom branch is the mask branch.

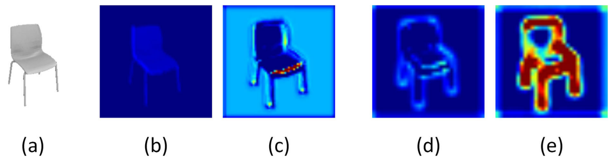


FIGURE 5. Visualization of features before and after attention modules. (a) The input image. (b) Features before passing to the first attention module. (c) Features after the first attention module. (d) and (e) Features before and after the second attention module. Notice that attention modules focus on and capture the edge and legs of the chair.

attention-based encoder aiming to focus on the specific trunk, branch structures or details of objects in images. The encoder is constructed by stacking multiple attention modules to obtain attention-aware image features. In order to make the network easier to train, we add residual connections inside and outside the attention modules. The encoder maps the input image to an embedding space and obtains feature z. Our decoder is constructed by a set of deconvs and predicts point clouds from image features. Compared with full connections, deconvs can combine features of the decoder with the corresponding features of the encoder through U-Net [34] which enhances features. Convs in image encoder and deconvs in point cloud decoder are symmetrical operations. Before the last FC layer in point cloud decoder, image features and point cloud features are consistent in size and channel in corresponding layers, which are both (B, H, W, C), so we could concat them directly. The decoder takes feature z as input and finally outputs a matrix of shape $1024 * 3$, where each row contains the coordinates of each point. The key difference between our work and existing 3d reconstruction works is the introduction of attention mechanism which is implemented by attention modules.

1) ATTENTION MODULE

As shown in Figure 4, the attention module is divided into two branches: trunk branch and attention mask branch. The trunk branch performs the function of a conventional encoder that extracts image features $T(x)$ through CNN. The attention mask branch learns a mask $M(x)$ which has the same size as $T(x)$ through an hourglass structure. The key to implementing an attention mechanism is the attention mask branch. Given input features, down samplings are performed several times to increase the receptive field and extract global features. Then the global features are extended to the original size by a set of symmetrical upsampling to obtain the mask. A sigmoid layer is followed to normalize the mask range to [0,1]. The mask can guide output features in each position after the dot product the mask and the output of trunk branch. Specifically, we use residual connection inside attention module so the output of attention module is:

$$H_i(x) = (1 + M_i(x)) * T_i(x) \tag{1}$$

where i denotes channel i of feature maps.

B. DENSE POINT CLOUD GENERATION

After generating sparse point clouds, we build a dense point cloud generation network to densify the generated sparse point cloud to a dense point cloud. Specifically, we generate a dense point cloud of 16384 points from a sparse point cloud of 1024 points through two dense modules without intermediate input and output. The dense module consists of feature extraction and feature expansion. A skip-connection enhances the communication between two dense modules through feature interpolation. The network structure is shown in Figure 6.

1) FEATURE EXTRACTION

We strive for extracting features from a generated sparse point cloud. In point cloud processing, PointNet [5] is very effective for extracting global features of point clouds and performs well in classification tasks. Thus, we adopt an MLP structure similar to PointNet to operate every individual point and obtain global features of shape $N * C_g$ via max-pooling the output of a set of MLPs. However, global features cannot represent local geometric information. DensePCR [30] and EC-Net [35] adopt Pointnet++ [6] to extract local features of point clouds. But Pointnet++ still treats individual points in local point sets independently by point set sampling and does not consider relationships between point pairs. As the number of points increasing, sampling and finding neighboring points will cost more memories and calculation. Inspired by dynamic graph convolution [26], we define local neighborhood in feature space and adopt a set of edge convolutions to extract local features. Given point coordinates of shape $N * 3$ from the input point cloud or point features of shape $N * C_f$ from a previous layer, we compute edge features for each point by applying an MLP and obtain a tensor of shape $N * C_e$ after max-pooling among neighboring edge features, where C_f is the channel of input point features and C_e is the number of neurons of the MLP. The local neighborhood is computed by KNN search in the feature space and updated

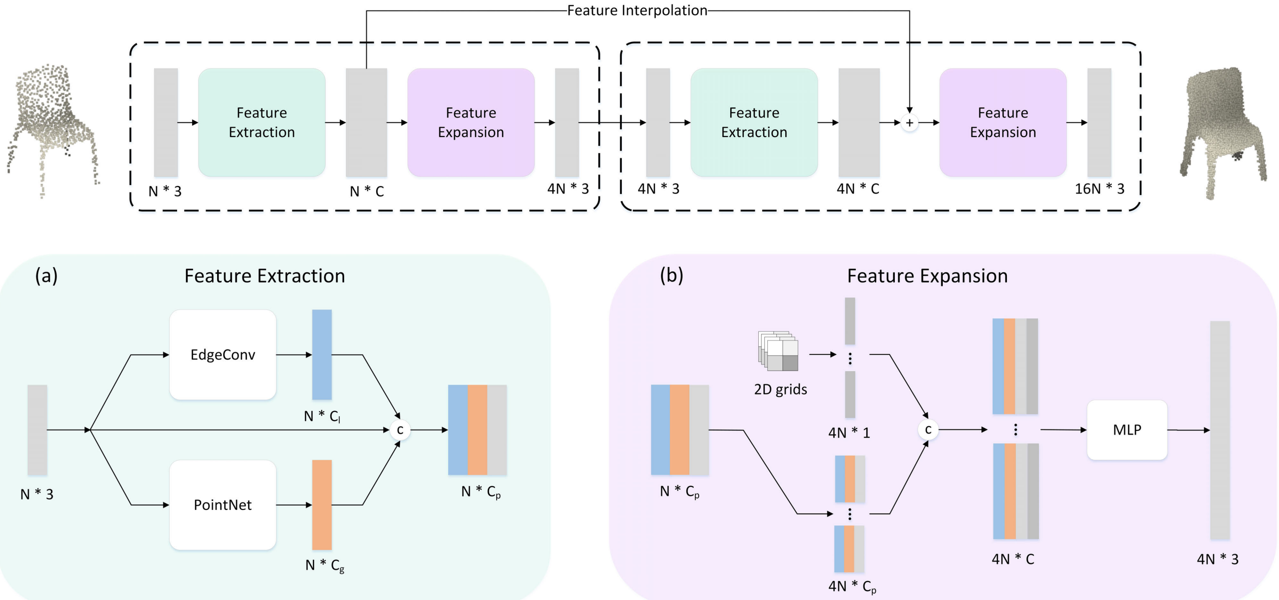


FIGURE 6. Dense point cloud generation network. The network consists of two consequent dense modules and each dense module contains feature extraction and feature expansion. (a) Feature extraction concatenates point feature ($N * 3$), global feature ($N * C_g$) and local feature ($N * C_l$) then outputs the concatenated feature ($N * C_p$) where $C_p = 3 + C_g + C_l$. (b) Feature Expansion concatenates the tiled concatenated feature ($4N * C_p$) and tiled grid feature ($4N * 1$) to obtain final feature ($4N * C$) where $C = C_p + 1$, then generate a denser point cloud of dimension $4N * 3$ through a set of MLPs. Feature interpolation enhances communication between two dense modules.

dynamically due to the different feature outputs of each layer. Then we concatenate point coordinates, global features, and local features to obtain the concatenated feature of shape $N * C_p$ and pass it to next stage as shown in Figure 6 (a).

2) FEATURE INTERPOLATION

While DensePCR trains two dense reconstruction networks independently, we only train one dense point cloud generation network by introducing a skip-connection to enhance the communication between dense modules. Due to different point cloud scales between two dense modules, we need to adopt feature interpolation while connecting features from two dense modules. Inspired by [22], we use bilateral interpolation. For the first dense module d_1, p_i and F_i denote the coordinates of i-th point and its extracted features respectively while $p_{i'}$ and $F_{i'}$ denote the coordinates of i'-th point and its extracted features respectively in the second dense module d_2. $N_{i'}$ denotes the spatial KNN of p_i from d_2. The interpolated feature $\widetilde{F}_i$ is:

$$\widetilde{F}_i = \frac{\sum_{i' \in N_{i'}} \theta\left(p_i, p_{i'}\right) \varphi\left(F_i, F_{i'}\right) F_{i'}}{\sum_{i' \in N_{i'}} \theta\left(p_i, p_{i'}\right) \varphi\left(F_i, F_{i'}\right)} \tag{2}$$

where θ and φ are two Gaussians defined as:

$$\theta\left(p_i, p_{i'}\right) = e^{-\left(\frac{\|p_i - p_i\|}{r}\right)^2} \tag{3}$$

$$\varphi\left(F_i, F_{i'}\right) = e^{-\left(\frac{\|F_i - F_i\|}{h}\right)^2} \tag{4}$$

The width parameters r and h denote the average distance to the closest neighbor.

Instead of concatenating interpolated features and extracted features which would widen the network, we apply a residual skip-connection, i.e., $\widetilde{F}_i = F_{i'} + F_i$.

3) FEATURE EXPANSION

In this stage, we expand features to an upsampled set of point coordinates and the upsampling factor is 4. PU-Net [29] and EC-Net [35] replicate each point for 4 times and pass each replicant independently to an individual set of MLPs, which may cause the replicant points to cluster around the original points. The repulsion loss is introduced to separate these clustered points but also brings more calculation and may mislead the distribution of points. Instead of applying MLPs to expand feature, we replicate extracted point features for 4 times to obtain expanded point features of shape $4N * C_p$ and assign N 2D grids to the replicated and overlapped points. These 2D grids learn to fit the surfaces near the points of a denser point cloud and guide replicated points to distribute around the surfaces. With the help of these 2D grids, the network learns to separate points without using the repulsion loss. We reshape these 2D grids of shape $2 * 2$ to $4 * 1$ where each 1D grid corresponds every single point and obtain grid features of shape $4N * 1$. After concatenating grid features and expanded point features, the final point features of shape $4N * C$ are passed through a set of MLPs to predict point coordinates of shape $4N * 3$ as shown in Figure 6 (b).

C. TRAINING STRATEGY

We adopt a two-stage training strategy by first training sparse point cloud generation network and dense point cloud generation network separately and then combining two networks by finetuning. Finally, we obtain an end-to-end network that takes a single image as input and outputs a dense point cloud.

We use the most commonly-used loss function in point cloud reconstruction tasks, chamfer distance [4], to train both of two stages. The CD is a point-wise L2 distance that measures the distance between two point sets. For each point in point set S_1, the CD finds the nearest neighbor in the other point set S_2 then sums the squared distances up, and vice versa. It is defined as:

$$d_{CD}\left(S_1, S_2\right)=\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2+\sum_{y\in S_2}\min_{x\in S_1}||x-y\|_2^2 \tag{5}$$

where x denotes points in point set S_1 and y denotes points in point set S_2.

The CD is continuous-differential and the computation of each point pair is independent so it is an efficient loss function. It can evaluate the overall similarity between two point sets but it only focuses on finding the nearest neighbor point pairs without checking that whether some points regard exactly the same point in the other point set as their nearest neighbor, which may make these points tend to cluster wrongly and lead to a point set uniformity problem. In this case, the CD metric may be low but the two-point sets may not be similar. The Earth Mover's distance [4] solves the problem by enforcing a point-to-point mapping between two point sets, which is defined as:

$$d_{EMD}\left(S_1, S_2\right) = \min_{\phi:S_1\rightarrow S_2}\sum_{x\in S_1}\|x-\phi(x)\|_2 \tag{6}$$

where ϕ is a bijection.

However, the EMD is computationally expensive as a loss function, especially for dense point clouds. In order to take advantages of both loss functions, we first train our two stages separately using the CD loss. When combining the two stages together and finetuning, we optimize the EMD loss of the first stage to obtain relatively uniform sparse point clouds. We also optimize the CD loss between the outputs of the first dense module and the ground truth.

IV. EXPERIMENTS

A. DATASET

Our 3D models come from ShapeNet [8] dataset. In order to compare with previous works, we select about 44k synthetic models from 13 different categories and use the 80%-20% train/test split provided by [2]. The input images provided by [2] and used in previous works [15], [30] are randomly rendered from 24 different azimuth angles which contain many bad viewpoints that can not express the overall shape of models in images. Besides, previous works only used one image of 24 of each model as their input and sampled points on mesh surfaces of the model as the ground truth but without aligning them with the image of the corresponding viewpoint. Without this image-to-model mapping, the network may be hard to train and the rest of 23 images are useless. PSGN [4] was trained with a translated, rotated, and scaled version of ShapeNet with parameters we do not have access to. Thus, we render each 3D model into RGB images with the resolution of 256 ∗ 256 from 12 fixed azimuth angles as our input images. We find some models with broken textures which may mislead network training so we remove the textures of models. We sample 1024 and 16384 points on the mesh surface of each model uniformly using farthest point sampling as the ground truth point clouds. The number of sampled points of the ground truth is selected to be able to compare with previous works. Then we copy and align each point cloud with the image of the corresponding viewpoint to build an image-to-model mapping for all 12 rendered images of each model, which gets full use of all rendered images and augments the dataset.

B. IMPLEMENTATION

Our network is trained in two stages. First, our attention-based sparse point cloud generation network starts with an encoder which consists of a set of convs, 3 attention modules with 3 residual blocks followed and final 3 residual blocks. The encoder takes an RGB image with the size of 256 ∗ 256 ∗ 3 as input and outputs image features of dimension 4 ∗ 4 ∗ 512. The decoder that consists of a set of convs and deconvs generates a sparse point cloud of 1024 points from the image features by predicting point coordinates of dimension 1024 ∗ 3. Then the generated sparse point cloud is passed through our dense point cloud generation network which consists of two consequent dense modules. Either dense module extracts global features using MLPs of dimension [32, 64, 64] and local features using edge convs of dimension [32, 64, 128] where we set k of nearest neighbors to be 20 in edge convs. Global features and local features are concatenated and then passed from the first dense module to the second dense module by feature interpolation. After tiling the concatenated features for 4 times, we assigned 2D grids with value [[−0.1,0.1], [−0.1,0.1]] to each point to expand features. The concatenated features are passed to MLPs of dimension [128, 128, 64, 3] which outputs 4 times denser point clouds. After passing through two consequent dense modules, the generated sparse point cloud of 1024 points is densified to a dense point cloud of 16384 points. We finally finetune the two stages to obtain an end-to-end network. We use Adam optimizer with a learning rate of 3e-5 and a minibatch size of 10.

C. EVALUATION METHODOLOGY

To evaluate quantitatively, we report the CD metric and the EMD metric introduced in Section III-C as previous works did. The CD metric efficiently evaluates the overall similarity between two point clouds and the EMD metric is more expressive in point cloud uniformity. More details of these two metrics can refer to [4]. The implementation of CD and EMD are followed by [4] and [36], which are also used in [15], [24], [30]. For CD and EMD, smaller is better.

TABLE 1. Comparison with PSGN-FC and DensePCR on ShapeNet dataset. CD and EMD are both scaled by 1e3. Smaller is better. Our approach is better than PSGN-FC and DensePCR in most categories on in most categories on both CD and EMD.

Category	CD			EMD		
	PSGN-FC	DensePCR	Ours	PSGN-FC	DensePCR	Ours
rifle	1.450	**1.048**	1.443	4.050	**1.295**	1.432
lamp	5.246	4.359	**3.465**	4.696	1.886	**1.836**
cabinet	0.778	0.909	**0.738**	4.453	1.240	**1.209**
telephone	1.157	0.925	**0.881**	3.767	**1.028**	1.069
bench	3.092	1.997	**1.233**	5.003	1.279	**1.109**
sofa	1.491	1.745	**1.085**	5.052	1.317	**1.176**
chair	3.591	2.630	**1.507**	4.661	1.485	**1.294**
airplane	3.446	**3.093**	4.232	3.945	**1.805**	2.050
table	3.559	1.695	**1.266**	4.762	1.236	**1.153**
monitor	1.748	1.629	**1.339**	4.951	1.361	**1.274**
car	**0.925**	1.069	0.931	4.965	1.280	**1.271**
vessel	**1.327**	1.601	1.603	4.479	**1.387**	1.439
speaker	1.251	1.398	**1.111**	4.759	1.318	**1.292**
mean	2.236	1.847	**1.603**	4.580	1.378	**1.354**

D. BASELINES

We consider the dense version of PSGN [4] and DensePCR [30] as the baselines for dense point cloud generation task. We implement both the dense version of PSGN and DensePCR following the implementations in DensePCR and train them on the same ShapeNet dataset to have a fair comparison. We select PSGN-FC with a decoder consists of fully connected layers to be the dense version of PSGN, which performs better than PSGN-ConvFC with a decoder consists of both deconvs and fully connected layers according to DensePCR. Since DensePCR is a multi-stage training network similar to ours, we also consider the sparse generation network of DensePCR which does not introduce attention mechanism as the baseline for sparse point cloud generation to highlight the efficacy of our attention-based encoder for sparse point cloud generation. In order to have a fair comparison and ensure that all networks have been converged, the sparse point cloud generation of DensePCR is trained 50 epochs and two dense reconstruction networks of DensePCR are trained 20 epochs. PSGN-FC and fine-tuning stage of DensePCR are trained 40 epochs. Our sparse point cloud generation network and dense point cloud generation network are trained 50 and 20 epochs respectively and finetuning stage is trained 40 epochs which are the same as DensePCR.

E. EVALUATION ON ShapeNet

We evaluate our approach on ShapeNet [8] dataset. Given a single RGB image as input, we first generate a sparse point cloud and then densify it to a dense point cloud. A sparse point cloud has 1024 points, corresponding to PSGN [4] that also reconstruct point clouds of 1024 points. A dense point cloud has 16384 points, corresponding to the ground truth point clouds of 16384 points that are commonly used in previous works [4], [14], [30].

We report the CD and EMD of our approach for all categories compared with PSGN-FC and DensePCR in Table 1. We outperform them in 9 out of 13 categories in CD while our mean score of CD is 28.3% lower than that of PSGN-FC and 13.2% lower than that of DensePCR. As for EMD, we outperform PSGN-FC in all categories and outperform DensePCR 9 out of 13 categories while our mean score of EMD is 70.4% lower than that of PSGN-FC and 1.7% lower than that of DensePCR.

TABLE 2. Comparison between DensePCR (sparse) and Ours (sparse) on ShapeNet dataset. The mean scores of CD and EMD are both scaled by 1e3. Smaller is better. Our sparse point cloud generation outperforms DensePCR in 11 out of 13 categories. Since sparse point cloud generation is not what we focus on, we briefly report the comparison of mean scores of CD and EMD to highlight the efficacy of our attention-based encoder.

Metric	CD	EMD
DensePCR(sparse)	2.754	1.615
Ours(sparse)	**2.243**	**1.468**

Compared with PSGN-FC, the gain in both metrics of ours especially the significant gain in EMD can be attributed to our two-stage training strategy, which is generating sparse point clouds from single images and then densifies them to dense point clouds. Our first stage is optimized via the EMD loss, which ensures the generated sparse point clouds to be uniform while the cost of calculation and memory can be acceptable due to the small number of generated points of the first stage. Instead of directly generating dense point clouds, this two-stage training strategy ensures that we can train our network based on finer initial reconstruction results from the first stage to avoid propagating and extending errors during the next stage of training.

Compared with DensePCR, the gain in both metrics of ours can be attributed to our better sparse point cloud reconstruction results from the first stage with the help of attention mechanism. To better highlight the efficacy of our attention-based encoder, Table 2 shows that our sparse point cloud generation outperforms sparse point cloud generation of DensePCR while our mean scores of CD and EMD are 18.6% and 9.1% lower. Qualitative results are shown in Figure 7. Our attention-based encoder makes the network pay more attention to the trunk structures of object such as trunk parts of chairs and lamps while the branch details of object are also

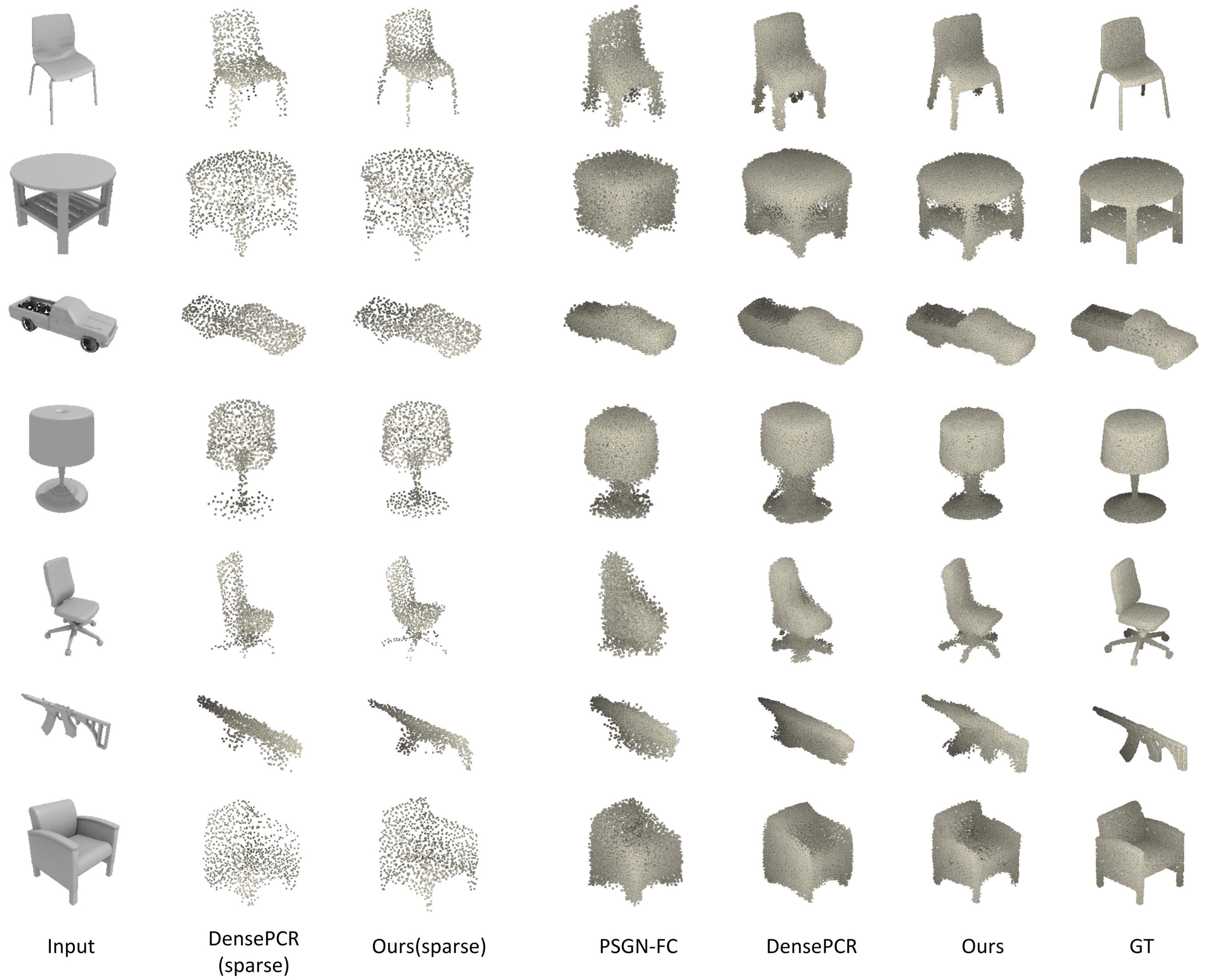


FIGURE 7. Qualitative results on ShapeNet. We compare our sparse point cloud generation with that of DensePCR at left. Each sparse point cloud has 1024 points. Notice that we capture finer details from the input images such as chairs' legs and the post of the lamp. We also compare our dense point cloud generation with PSGN-FC and DensePCR at right. Each dense point cloud has 16384 points We preserve both the overall shapes and details after densifying sparse point clouds.

preserved such as chair legs. The encoder of DensePCR does not recognize the thin and tight structures so it might miss the structure details or cluster more points on them. Notice that predicted points of our sparse point generation distribute more exactly and reasonably, which would not expand thin and tight structures like chair legs to thick or sparse ones while still maintain the high quality of the overall shapes. Besides, we need to point out that our dense point cloud generation does not take extra point clouds as intermediate input in training, while DensePCR needs to firstly densify sparse point clouds with 1024 points to 4096 points and then to 16384 points so it needs extra intermediate ground truth point clouds of 4096 points and needs to train two dense reconstruction networks. Our approach directly predicts dense point cloud of 16384 points from sparse point clouds of 1024 points in one shot. Even without extra data and extra training, we still obtain better EMD metric than DensePCR while significantly outperform it in CD while our dense point cloud reconstruction results preserve both overall shapes and branch details.

But failure cases still exist in our experiment. As shown in Table 1, we reconstruct chairs, tables, benches, and sofas much better than PSGN-FC and DensePCR. These categories share some common features such as main large plane surfaces, tight and long branches, clear boundaries and edges. Due to a large amount of these categories in training data, our attention mechanism captures these features well. But airplanes do not share these features while they have much thinner and slighter wings and empennages. Due to the normalization of 3D models which ensures models of each category are on the same scale, airplanes become relatively smaller with even smaller and unclear details such as engines

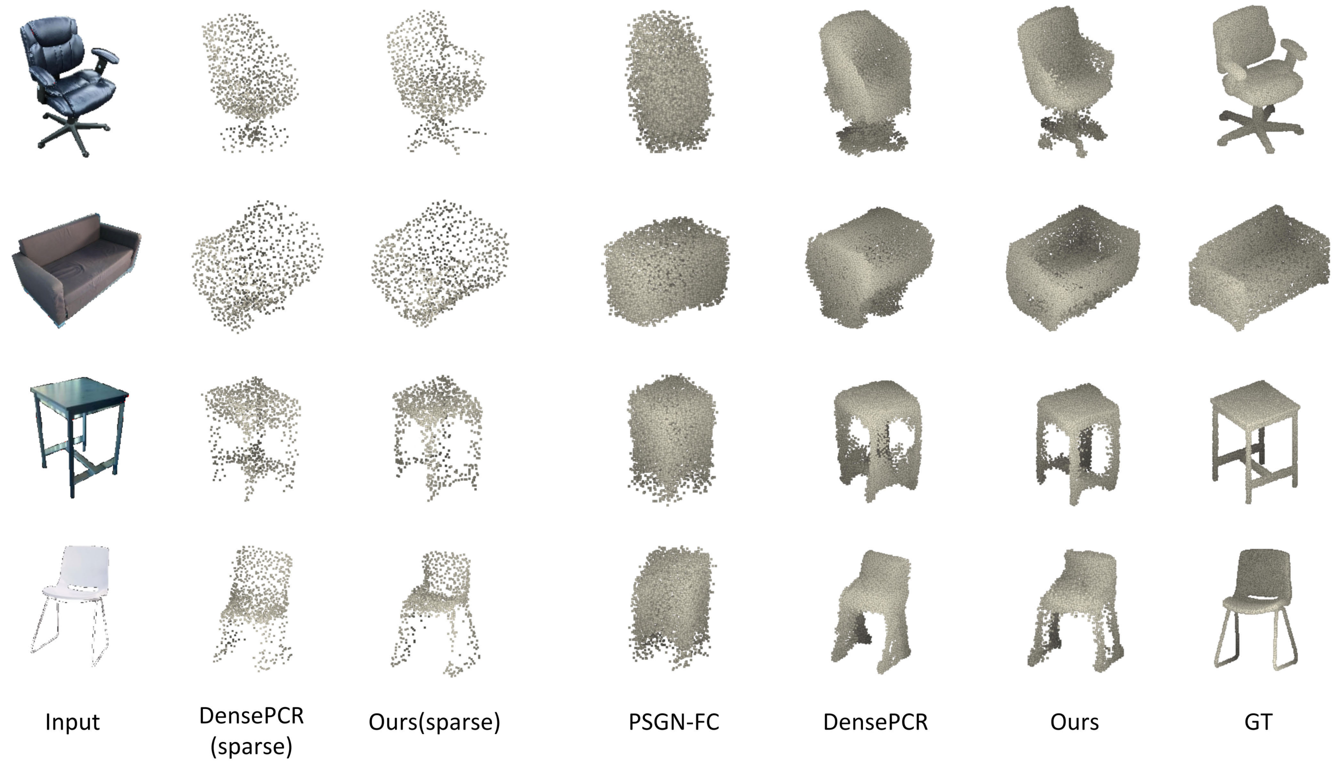


FIGURE 8. Qualitative results on Pix3D. We compare our sparse point cloud generation with that of DensePCR at left and our dense point cloud generation with PSGN-FC and DensePCR at right. We generate more resonable point clouds even with the input from a different distribution. All models are trained on ShapeNet training set and test on Pix3D.

and empennages in rendered images. Besides, the cabin and wings of airplanes lie in the horizontal direction. They might become shorter in a large number of images because the viewpoint also rotates horizontally around the center of objects. Without depth information, the network can not estimate the length of wings in some images at certain viewpoints. These factors make our attention mechanism less effective in capturing features in airplanes. But our approach still obtains better results in most categories and mean scores.

We also perform an ablation study to demonstrate the efficacy and contribution of our two important components: attention modules and feature interpolation. Table 3 reports the performance of each network by removing one component from the full network. We first remove attention modules and corresponding residual connections from our sparse point cloud generation network. The mean scores of CD and EMD are 1.4% and 8.2% higher than our full network. We then remove feature interpolation from our dense point cloud generation network. Without the feature interpolation and skip connection between two dense modules, the mean scores of CD and EMD are 4.2% and 0.6% higher than our full network. Both these two components contribute to the performance of our final network.

TABLE 3. Ablation study that evalutates the contribution of each component to the performance of the network.

Metric	CD	EMD
- attention	1.625	1.465
- feature interpolation	1.671	1.362
full	**1.603**	**1.354**

F. RECONSTRUCTION FROM REAL-WORLD IMAGES

We test our network on Pix3D [9] dataset which consists of real-world images and corresponding masks, poses and ground truth 3d models. We evaluate our trained model on categories that cooccur in ShapeNet dataset and exclude images having occlusion and truncation from the test set as previous works [9], [30] did. We mask the backgrounds in images using the provided masks and move the main objects to the center position and resize the images to $256 * 256$ as our input. We need to point out that real-world images of real objects differ from rendered images of synthetic models in two aspects. First, illumination in the real world may bring many changes to photos of objects such as brightness and shadows. Second, details of objects in real world are more than synthetic models. But our approach still obtains acceptable reconstruction results from real-world images. We also compare our approach with baselines. All of our model and baselines' models are trained on ShapeNet dataset.

As shown in Table 4, our approach outperforms DensePCR in all categories while our mean scores of CD and EMD are 12.4% and 0.7% lower respectively. Our approach also outperforms the dense version of PSGN-FC significantly in EMD with a 50.5% lower mean score but obtain a higher mean score on CD. Figure 8 shows some sample visualization results. We notice that even with the best CD metric,

TABLE 4. Comparison with PSGN-FC and DensePCR on Pix3D dataset. CD and EMD are both scaled by 1e3. Smaller is better.

Category	CD			EMD		
	PSGN-FC	DensePCR	Ours	PSGN-FC	DensePCR	Ours
table	15.777	16.076	**14.053**	5.526	2.869	**2.849**
sofa	**5.215**	8.084	6.962	5.343	2.313	**2.292**
chair	**10.661**	13.721	12.151	4.740	2.600	**2.585**
mean	**10.551**	12.627	11.055	5.203	2.594	**2.576**

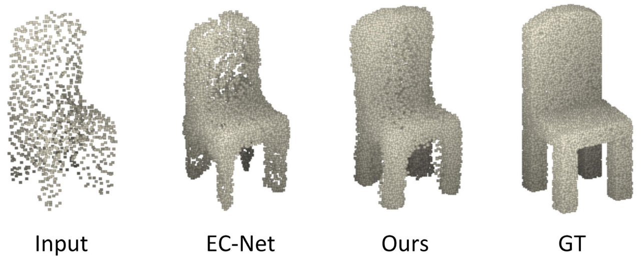


FIGURE 9. Comparison to EC-Net. Given the input generated sparse point cloud, the dense point cloud generated by EC-Net leaves many holes while ours fills them in and keeps the point cloud tight.

PSGN-FC gets non-ideal reconstruction results and predicts highly incoherent point clouds with many clustered points. As we have analyzed the CD in Section III-C, the limitation of the CD is that the metric can be low when both predicted and ground truth point clouds consist of a large number of clustered points because it is not a point-to-point mapping. Point clouds of sofas and many chairs consist of many clustered points. So the CD may not evaluate the reconstruction quality precisely especially when the EMD is not so good. Our predicted point clouds are corresponded with input images and still preserve overall shapes and capture more shape details such as legs of chairs and tables or edges of sofas than point clouds generated by PSGN-FC and DensePCR.

G. COMPARISON TO POINT CLOUD UPSAMPLING METHOD

Although there are some works directly focus on point cloud upsampling such as PU-Net [29] and EC-Net [35], our work is different from them. EC-Net, the improved version of PU-Net, is designed to upsample highly uniform point clouds. It is trained by points grouped in local patches and tries to learn local information of point clouds. Its training is easy and efficient because it only processes a small number of points, which would not cost much calculation and memory when computing the CD or other losses and finding nearest neighbors. It performs well in highly uniform point clouds but can not handle point clouds generated from images, which leads to the point cloud uniformity problem. The generated point clouds are not highly uniform even though the network is trained with EMD loss, which means there might be some sparse parts of the object where it is supposed to be dense. EC-Net only cares about local information so it might remain holes after upsampling input point clouds, which is shown as Figure 9. But our work can handle the predicted point clouds and fill the holes while upsampling point clouds by learning both global and local features of point clouds.

V. DISCUSSION

Our method and most existed works on 3D reconstruction reconstruct single objects from images without backgrounds. Though we can reconstruct 3D models from images of indoor or outdoor scenes, the backgrounds of images must be masked as we mentioned in Section IV-F. Zhang *et al.* [16] was able to reconstruct 3D models from images of complex backgrounds by combining features of images and nearest-shape retrievals from the synthetic dataset. It is a novel trying and works well for images of complex backgrounds. But retrievals of nearest-shape are still not exactly the shapes of input images, which means there might be some native errors from the beginning of reconstruction. Reconstructing objects from images of complex backgrounds may be an open research interest. Training an image segmentation network to capture main objects from complex backgrounds and feeding images into the image segmentation network before passing them into reconstruction network may be a possible solution.

VI. CONCLUSION

We propose an attention-based dense point cloud generation network which takes a single RGB image as input and generates a dense point cloud. We introduce an encoder based on the attention mechanism to point cloud reconstruction task. We first generate a sparse point cloud from a single image and then densify it to a dense point cloud through our dense point cloud generation network. Our evaluation on synthetic and real-world datasets shows that our approach generates high-quality dense point clouds from single images and is robust to handle a new and unseen dataset. In the future, we need to improve our attention mechanism to fit more categories of objects. Besides, our approach still predicts some discrete points around the outlines of generated point clouds, which may be improved by optimizing the process of point cloud processing.

REFERENCES

[1] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 628–644.

[3] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 484–499.

[4] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 605–613.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 652–660.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," Dec. 2015, *arXiv:1512.03012*. [Online]. Available: https://arxiv.org/abs/1512.03012

[9] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman, "Pix3D: Dataset and methods for single-image 3D shape modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2974–2983.

[10] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2088–2096.

[11] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, p. 72, 2017.

[12] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 412–420.

[13] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese, "Deformnet: Free-form deformation network for 3D shape reconstruction from a single image," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 858–866.

[14] D. Jack, J. K. Pontes, S. Sridharan, C. Fookes, S. Shirazi, F. Maire, and A. Eriksson, "Learning free-form deformations for 3D object reconstruction," Mar. 2018, *arXiv:1803.10932*. [Online]. Available: https://arxiv.org/abs/1803.10932

[15] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu, "3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image," Jul. 2018, *arXiv:1807.07796*. [Online]. Available: https://arxiv.org/abs/1807.07796

[16] Y. Zhang, Z. Liu, T. Liu, B. Peng, and X. Li, "Realpoint3D: An efficient generation network for 3D object reconstruction from a single image," *IEEE Access*, vol. 7, pp. 57539–57549, 2019.

[17] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[18] K. Li, T. Pham, H. Zhan, and I. Reid, "Efficient dense point cloud object reconstruction using deformation vector fields," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 497–513.

[19] K. L. Navaneet, P. Mandikal, M. Agarwal, and R. V. Babu, "CAPNet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision," Nov. 2018, *arXiv:1811.11731*. [Online]. Available: https://arxiv.org/abs/1811.11731

[20] L. Jiang, S. Shi, X. Qi, and J. Jia, "GAL: Geometric adversarial loss for single-view 3D-object reconstruction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 802–816.

[21] R. Sun, Y. Gao, Z. Fang, A. Wang, and C. Zhong, "SSL-Net: Point-cloud generation network with self-supervised learning," *IEEE Access*, vol. 7, pp. 82206–82217, 2019.

[22] W. Yifan, S. Wu, H. Huang, D. Cohen-Or, and O. Sorkine-Hornung, "Patch-based progressive 3D point set upsampling," Nov. 2018, *arXiv: 1811.11286*. [Online]. Available: https://arxiv.org/abs/1811.11286

[23] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3D surface generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.

[24] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 52–67.

[25] J. Pan, J. Li, X. Han, and K. Jia, "Residual MeshNet: Learning to deform meshes for single-view 3D reconstruction," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2018, pp. 719–727.

[26] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," Jan. 2018, *arXiv:1801.07829*. [Online]. Available: https://arxiv.org/abs/1801.07829

[27] J. Li, B. M. Chen, and G. H. Lee, "So-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[28] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," Nov. 2018, *arXiv:1811.07246*. [Online]. Available: https://arxiv.org/abs/1811.07246

[29] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-Net: Point cloud upsampling network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799.

[30] P. Mandikal and V. B. Radhakrishnan, "Dense 3D point cloud reconstruction using a deep pyramid network," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1052–1060.

[31] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 483–499.

[32] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3156–3164.

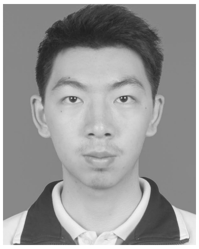
[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Springer, 2015, pp. 234–241.

[35] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Ec-Net: An edge-aware point set consolidation network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 386–402.

[36] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," Jul. 2017, *arXiv:1707.02392*. [Online]. Available: https://arxiv.org/abs/1707.02392

QIANG LU received the master's and Ph.D. degrees in computer science and information from the Hefei University of Technology. He was a Visiting Scholar with the University of North Texas. He is currently an Associate Professor with the School of Computer and Information, Hefei University of Technology, where he is a member of the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education. His primary research interests include visualization, computer graphics, and cooperative computing. He is a member of the CCF.



MINGJIE XIAO was born in China, in 1995. He received the B.S. degree from the Hefei University of Technology (HFUT), Hefei, China, in 2018, where he is currently pursuing the master's degree with the School of Computer and Information. His research interests include 3D reconstruction, computer vision, and machine learning.

YIYANG LU was born in China, in 1996. He received the B.S. degree from the Hefei University of Technology (HFUT), Hefei, China, in 2018, where he is currently pursuing the master's degree with the School of Computer and Information. His research interests include 3D reconstruction, computer vision, and machine learning.

XIAOHUI YUAN (S'01–M'05–SM'16) received the B.S. degree in electrical engineering from the Hefei University of Technology, China, in 1996, and the Ph.D. degree in computer science from Tulane University, in 2004. He is currently an Associate Professor with the University of North Texas. His research findings have been published in more than 140 peer-reviewed articles. His research interests include computer vision, artificial intelligence, data mining, and machine learning. He was a recipient of the Ralph E. Powe Junior Faculty Enhancement Award, in 2008. He is the Chair of several international conferences. He is the Editor-in-Chief of the *International Journal of Smart Sensor Technologies and Applications* and serves on the Editorial Board of several international journals.

YE YU was born in Anqing, Anhui, China, in 1982. She received the Ph.D. degree in computer science and technology from the Hefei University of Technology (HFUT), in 2010, where she is currently an Associate Professor with the School of Computer and Information. Her current research interests include computer vision, object detection and classification, 3D modeling, and virtual reality.