L2G Auto-encoder: Understanding Point Clouds by Local-to-Global Reconstruction with Hierarchical Self-Attention

Xinhai Liu
School of Software, Tsinghua University & Beijing National Research Center for Information Science and Technology (BNRist)
Beijing, China
lxh17@mails.tsinghua.edu.cn

Zhizhong Han
Department of Computer Science, University of Maryland
College Park, USA
h312h@umd.edu

Xin Wen
School of Software, Tsinghua University & Beijing National Research Center for Information Science and Technology (BNRist)
Beijing, China
x-wen16@mails.tsinghua.edu.cn

Yu-Shen Liu*
School of Software, Tsinghua University & Beijing National Research Center for Information Science and Technology (BNRist)
Beijing, China
liuyushen@tsinghua.edu.cn

Matthias Zwicker
Department of Computer Science, University of Maryland
College Park, USA
zwicker@cs.umd.edu

ABSTRACT

Auto-encoder is an important architecture to understand point clouds in an encoding and decoding procedure of self reconstruction. Current auto-encoder mainly focuses on the learning of global structure by global shape reconstruction, while ignoring the learning of local structures. To resolve this issue, we propose Local-to-Global auto-encoder (L2G-AE) to simultaneously learn the local and global structure of point clouds by local to global reconstruction. Specifically, L2G-AE employs an encoder to encode the geometry information of multiple scales in a local region at the same time. In addition, we introduce a novel hierarchical self-attention mechanism to highlight the important points, scales and regions at different levels in the information aggregation of the encoder. Simultaneously, L2G-AE employs a recurrent neural network (RNN) as decoder to reconstruct a sequence of scales in a local region, based on which the global point cloud is incrementally reconstructed. Our outperforming results in shape classification, retrieval and upsampling show that L2G-AE can understand point clouds better than state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Shape representations**; • **Information systems** → *Information retrieval.*

KEYWORDS

auto-encoder; unsupervised learning; hierarchical attention; interpolation layer; recurrent neural network

ACM Reference Format:
Xinhai Liu, Zhizhong Han, Xin Wen, Yu-Shen Liu, and Matthias Zwicker. 2019. L2G Auto-encoder: Understanding Point Clouds by Local-to-Global Reconstruction with Hierarchical Self-Attention. In *Proceedings of the 27th ACM International Conference on Multimedia (MM'19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350960

*Corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350960

1 INTRODUCTION

In recent years, point clouds have attracted increasing attention due to the popularity of various depth sensors in different applications. Not only the traditional methods, deep neural networks have also been applied to point cloud analysis and understanding. However, it remains a challenge to directly learn from point clouds. Different from 2D images, point cloud is an irregular 3D data which makes it difficult to directly use traditional deep learning framework, e.g., traditional convolution neural network (CNN). The traditional CNN usually requires some fixed spatial distribution around each pixel so as to facilitate the convolution. One way to alleviate the problem is to voxelize a point cloud into voxels and then apply 3D Cov-Nets. However, because of the sparsity of point clouds, it leads to resolution-loss and explosive computation complexity, which sacrifices the representation accuracy.

To address above challenges, PointNet [28] has been proposed to directly learn shape representations from raw point sets. Along with the availability of directly learning from point clouds by deep learning models, auto-encoder (AE) has become an vital architecture of the involved neural networks. Current AE focuses on the learning of the global structure of point clouds in the encoding and decoding procedure. However, current AE structure is still limited by learning the local structure of point clouds, which tends to be an important piece of information for point cloud understanding.

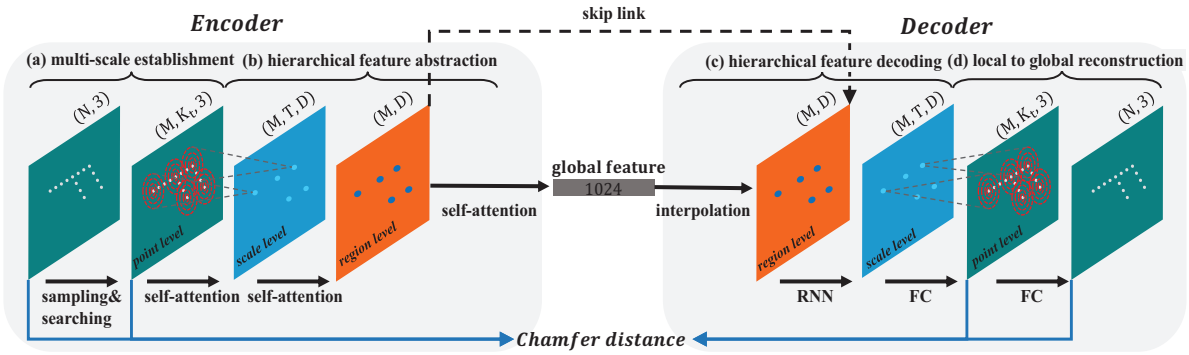

Figure 1: Illustration of our local to global auto-encoder architecture. In the encoder, multi-scale areas is established in each local region around the sampled centroids in (a). And a hierarchical feature abstraction is employed to abstract the global feature of point clouds with self-attention in (b). The learned global feature is applied to shape classfication and retrieval applications. In the decoder, local areas and the global point cloud are reconstructed by hierarchical feature decoding with the interpolation layer, the RNN layer and the FC layer in (c)(d).

To simultaneously learn global and local structure of point clouds, we propose a novel auto-encoder called Local-to-Global auto-encoder (L2G-AE). Different from traditional auto-encoder, L2G-AE leverages a local region reconstruction to learn the local structure of a point cloud, based on which the global shape is incrementally reconstructed for the learning of the global structure. Specifically, the encoder of L2G-AE can hierarchically encode the information at point, scale and region levels, where a novel hierarchical self-attention is introduced to highlight the important elements in each level. The encoder further aggregates all the information extracted from the point cloud into a global feature. In addition, L2G-AE employs a RNN-based decoder to decode the learned global feature into a sequence of scales in each local region. And based on scale features, the global point cloud is incrementally reconstructed. L2G-AE leverages this local to global reconstruction to facilitate the point cloud understanding, which finally enables local and global reconstruction losses to train L2G-AE.

Our key contributions are summarized as follows.

- We propose L2G-AE to enable the learning of global and local structures of point clouds in an auto-encoder architecture, where the local structure is very important in learning highly discriminative representations of point clouds.
- We propose hierarchical self-attention to highlight important elements in point, scale and region levels by learning the correlations among the elements in the same level.
- We introduce RNN as decoding layer in an auto-encoder architecture to employ more detailed self supervision, where the RNN takes the advantage of the ordered multi-scale areas in each local region.

2 RELATED WORK

Point clouds is a fundamental type of 3D data format which is very close to the raw data of various 3D sensors. Recently, applications of learning directly on point clouds have received extensive attention, including shape completion [33], autonomous driving [27], 3D object detection [32, 39, 47], recognition and classification [5, 23, 24, 28, 29, 31, 35, 37, 38, 42], scene labeling [22], upsampling [41, 44], dense labeling and segmentation [34] , etc.

Due to the irregular property of point cloud and the inspiring performances of 2D CNNs on large-scale image repositories such as ImageNet [4], it is intuitive to rasterize point clouds into 3D voxels and then apply 3D CNNs. Some studies [7, 27, 47] represent each voxel with a binary value which indicates the occupation of this location in space. The main problem of voxel-based methods is the fast growth of neural network size and computation complexity with the increasing of spatial resolution. To alleviate this problem, some improvements [25] have been proposed to explore the data sparsity of point clouds. However, when dealing with point clouds with huge number of points, the complexity of the neural network is still unacceptable.

Recently, deep neural networks work quite effectively on the raw 3D point clouds. Different from learning from readered views [6, 12–15, 17] 2D meshes [8] or 3D voxels [9–11], PointNet [28] is the pioneer study which directly learns the representation for point clouds by computing features for each point individually and aggregating these features with max-pool operation. To capture the contextual information of local patterns inside point clouds, PointNet++ [29] uses sampling and grouping operations to extract features from point clusters hierarchically. Similarly, several recent studies [21, 30] explores indexing structures, which divides the input point cloud into leaves, and then aggregates node features from leaves to the root. Inspired by the convolution operation, recent methods [24, 35, 38] investigate well-designed CNN-like operations to aggregate points in local regions by building local connections with k-neareat-neighbors (kNN).

Capturing the context information inside local regions is very important for the discriminative ability of the learned point cloud representations. KC-Net [31] employs a kernel correlation layer and a graph pooling layer to capture the local patterns of point clouds. ShapeContextNet [37] extends 2D Shape Context [2] to the 3D, which divides a local region into small bins and aggregates the bin features. Point2Seqeuce [26] employs an attention-based

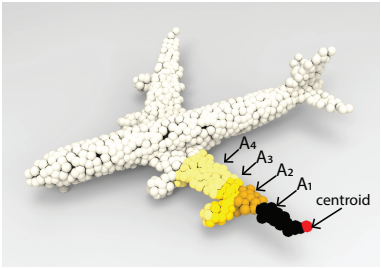

Figure 2: A multi-scale example inside a local region of an airplane point cloud, where there are four scales areas $[A_1, A_2, A_3, A_4]$ with different colors around the centroid point (red).

sequence to sequence architecture to encode the multi-scale area features inside local regions.

In order to alleviate the dependence on the labeled data, some studies have performed unsupervised learning for point clouds. FoldingNet [40] proposes a folding operation to deform a canonical 2D grid onto the suface of a point cloud. 3D-PointCapsNet [46] employs a dynamic routing scheme in the reconstruction of input point clouds. However, it is difficult for these methods to capture the local patterns of point clouds. Similar to FoldingNet, PPF-FoldNet [3] also learns local descriptors on point cloud with a folding operation. LGAN [1] proposes an auto-encoder based on PointNet and extends the decoder module to the point cloud generation application with GAN. In this work, we propose a novel auto-encoder architecture to learn representations for point clouds. On the encoder side, an hierarchical self-attention mechanism is applied to embedding the correlation among features in each level. And on the decoder side, an interpolation layer and a RNN decoding layer are engaged to reconstruct multi-scale areas inside local regions. After building local areas, the global point cloud is generated by a fully-connected (FC) layer which acts as a down sampling function.

3 METHOD

Now we introduce the L2G-AE in detail, where the structure is illustrated in Figure 1. The input of the encoder is an unordered point set $\boldsymbol{P} = \{p_1, p_2, \cdots, p_N\}$ with N ($N = 1024$) points. Each point in the point set is composed of a 3D coordinate (x, y, z). L2G-AE first establishes multi-scale areas $\boldsymbol{A_t}$ ($t \in [1, T]$) in each local region around the sampled points. Then, a hierarchical feature abstraction is enforced to obtain the global features of input point clouds with self-attentions. In the decoder, we simultaneously reconstruct local scale areas and global point clouds by hierarchical feature decoding. The output of L2G-AE is the reconstructed local areas $\boldsymbol{A}_t^{'}$ and the reconstructed $\boldsymbol{P}^{'}$ with same number of points to $\boldsymbol{P}$.

3.1 Multi-scale Establishment

To capture fine-grained local patterns of point clouds, we first establish multi-scale areas in each local region, which is similar to PointNet++ [29] and Point2Sequence [26]. Firstly, a subset $\{p_{i_1}, p_{i_2}, \cdots, p_{i_M}\}$ of the input points is selected as the centroid of local regions by iterative farthest point sampling (FPS). The latest point p_{i_j} is always the farthest one from the rest points $\{p_{i_1}, p_{i_2}, \cdots, p_{i_{j-1}}\}$. Compared to other sampling method, such as random sampling, FPS can achieve a better coverage of the entire point cloud with the given same number of centroids. As shown in Figure 2, around each sampled centroid, T different scale local areas are established continuously by kNN searching with $\{K_1, K_2, \cdots, K_T\}$ nearest points, respectively. An alternative searching method is ball query [29] which selects all points with a radius around the centroid. However, it is difficult for ball query to ensure the information inside local regions, which is sensitive to the sparsity of the input point clouds.

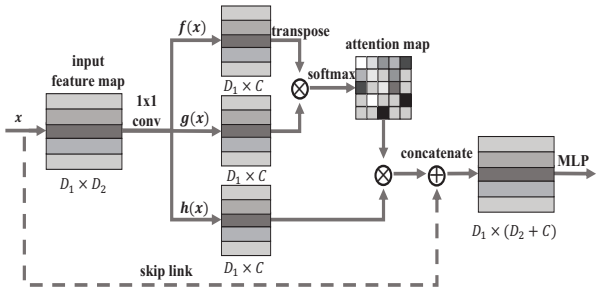

Figure 3: Self-attention module. The input of this module is a $D_1 \times D_2$ feature map and the output is another $D_1 \times (D_2 + C)$ feature map, where C is a parameter.

3.2 Hierarchical Self-attention

In current work of learning on point clouds, Multi-Layer-Perceptron (MLP) layer is widely applied to integrate multiple features. Traditional MLP layer first abstracts each feature into higher dimension individually and then aggregates these features by a concise max pooling operation. However, these two simple operations can hardly encode the correlation between feature vectors in the feature space. Inspired by the self-attention machanism in [45], the attention machanism is suitable for improving the traditional MLP by learning the correlation between features. In this work, we propose a self-attention module to make up the defects of the MLP layer with an attention mechanism. Here, self-attention refers to learn the correlation among features in the same level.

Different from the raw self-attention, we enforce a hierarchical feature extraction architecture with hierarchical self-attention in the encoder. There are three different levels inside the encoder, including point level, scale level, and region level. At each level, we introduce a self-attention module to learn self-attention weights by mining the correlations among the corresponding feature elements. Consequently, three self-attention modules are designed to propagate features from the lower level to the higher level. Supposed the input of the self-attention module is a feature map $\boldsymbol{x} \in \mathbb{R}^{D_1 \times D_2}$, where D_1, D_2 are the dimensions of the feature map. Therefore, D_1, D_2 are equal to K_t, 3 in the point level, equal to T, D in the scale level and equal to M, D in the region level, respectively.

As depicted in Figure 3, the feature map $\boldsymbol{x}$ is first transformed into two feature spaces $\boldsymbol{f}$ and $\boldsymbol{g}$ to calculate the attention below, where $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f}\boldsymbol{x}$, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{W_g}\boldsymbol{x}$,

$$\beta_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{D_1} exp(s_{ij})}, \text{where } s_{ij} = \boldsymbol{f}(\boldsymbol{x_i})^T \boldsymbol{g}(\boldsymbol{x_j}), \tag{1}$$

and $\beta_{j,i}$ evaluates the attention degree which the model pays to the i^{th} location when synthesizing the j^{th} feature vector. Then the attention result is $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r_j}, \cdots, \boldsymbol{r}_{D_1}) \in \mathbb{R}^{D_1 \times D_2}$, where

$$\boldsymbol{r_j} = \sum_{i=1}^{D_1} \beta_{j,i} \boldsymbol{h}(\boldsymbol{x_i}), \text{where } \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{W_h}\boldsymbol{x_i}. \tag{2}$$

In above formulation, $\boldsymbol{W_f}, \boldsymbol{W_g}, \boldsymbol{W_h} \in \mathbb{R}^{D_2 \times C}$ are learned weight matrices, which are implemented as 1×1 convolutions. We use $C = M/8$ in the experiments.

In addition, inspired by the skip link operation in ResNet[18] and DenseNet [20], we further concatenate the result of the attention mechanism with the input feature matrix. Therefore, the final output of the self-attention module is given by

$$\boldsymbol{o_i} = \boldsymbol{x_i} \oplus \boldsymbol{r_i}, \tag{3}$$

where $\oplus$ is the concatenation operation. This allows the network to rely on the cues among the feature vectors.

To aggregate the features with correlation information, a MLP layer and a max pooling operation are employed to integrate the multiple features. In particular, the first self-attention module aggregates the points in a scale to a D-dimensional feature vector. The second one encodes the multi-scale features in a region into a D-dimensional feature. The final one integrates features of all local regions on a point cloud into a 1024-dimensional global feature. Therefore, the encoder hierarchically abstracts point features from the levels of point, scale and region to a global representation of the input point cloud.

3.3 Interpolation Layer

The target of the decoder is to generate the points of the local areas and entire points. Previous approaches [1, 3, 40] usually use simple fully-connected (FC) layers or MLP layers to build the decoder. However, the expressive ability of the decoder is largely limited without considering the relationship among features. In this work, we propose a progressive decoding way which can be regarded as a reverse process of the encoding. The first step is to generate local region features from the global feature. To propagate the global feature $\boldsymbol{g}$ to region features, a simple interpolation operation is first engaged in the decoder. The local region feature $\boldsymbol{l}_i$ is calculated by

$$\boldsymbol{l}_i = \frac{c}{(p_i - p_0)^2} \boldsymbol{g}, i \in [1, M], \tag{4}$$

where c ($c = 10^{-10}$) is a constant. Here, $p_0 = (0, 0, 0)$ is the centroid of the input point cloud after the normalization processing. And p_i is the centroid point of the corresponding local region. By the simple interpolation operation, the spatial distribution information of local region can be integrated to facilitate the feature decoding. The interpolated local region features are then concatenated with skip linked local region features from the encoder. The concatenated features are passed through another MLP layer into a $M \times D$ feature matrix.

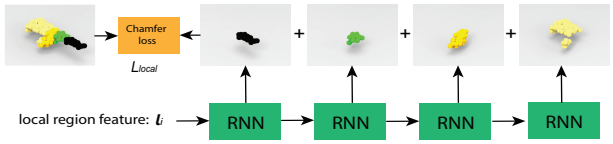

Figure 4: The decoding process of the RNN layer.

3.4 RNN Layer

Given the feature of local regions, we want to decode the scale level features. Due to the multi-scale setting, the features of different scales in a local region can be regarded as a feature sequence with length T. As we all know that recurrent neural network [19] has shown excellent performances in processing sequential data. Thus, a RNN decoding layer is employed to generate the multi-scale area features. The decoding process is shown in Figure 4. We first replicate the local region feature $\boldsymbol{l}_i$ for T times, and the replicated local region features are feed into the RNN layer by

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{l}_i^t), t \in [1, T], \tag{5}$$

where f is a non-linear activation function and t is the index of RNN step. Therefore, the predicted t^{th} area feature $\boldsymbol{a}_t$ can be calculated by

$$\boldsymbol{a}_t = \boldsymbol{W}_\theta \boldsymbol{h}_t. \tag{6}$$

Here, $\boldsymbol{W}_d$ is a learnable weight matrix. To generate the points inside each local area, several FC layers are adopted to reconstruct the points. The local area $\boldsymbol{A}_t^{'}$ is reconstructed by

$$\boldsymbol{A}_t^{'} = \boldsymbol{W}_{\theta_t} \boldsymbol{a}_t + b_{\theta_t}, \tag{7}$$

where $\boldsymbol{W}_{\theta_t}$, b_{θ_t} are weights of the FC layer. Based on the reconstructed local areas, another FC layer is applied to incrementally reconstruct the entire point cloud. All reconstructed areas are concatenated and then passed through the FC layer by

$$\boldsymbol{P} = \boldsymbol{W}[\boldsymbol{A}_1^{'} \oplus \boldsymbol{A}_2^{'} \oplus \cdots \oplus \boldsymbol{A}_T^{'}] + b. \tag{8}$$

Here, $\oplus$ represents the concatenation operation.

3.5 Loss Function

We propose a new loss function to train the network in an end-to-end fashion. There are two parts in the loss function, local scale reconstruction and global point cloud reconstruction, respectively. As mentioned earlier, we should encourage accurate reconstruction of local areas and the global point cloud at the same time. Suppose $\boldsymbol{A}_t$ is the t^{th} scale area in the multi-scale establishment subsection, then, the local reconstruction error for $\boldsymbol{A}_t^{'}$ is measured by the well-known Chamfer distance,

$$\begin{aligned} L_{local} = d_{CH}(\boldsymbol{A}_t, \boldsymbol{A}_t^{'}) = \sum_{t=1}^{T} (& \frac{1}{|\boldsymbol{A}_t|} \sum_{p_i \in \boldsymbol{A}_t} \min_{p_i^{'} \in \boldsymbol{A}_t^{'}} \|p_i - p_i^{'}\|_2 \\ & + \frac{1}{|\boldsymbol{A}_t^{'}|} \sum_{p_i^{'} \in \boldsymbol{A}_t^{'}} \min_{p_i \in \boldsymbol{A}_t} \|p_i - p_i^{'}\|_2), \end{aligned} \tag{9}$$

Similarly, let the input point set be $\boldsymbol{P}$ and the reconstructed point set be $\boldsymbol{P}^{'}$. The global reconstruction error can be denoted by

$$L_{global} = d_{CH}(\boldsymbol{P}, \boldsymbol{P}^{'}) = \frac{1}{|\boldsymbol{P}|} \sum_{p_i \in \boldsymbol{P}} \min_{p_i^{'} \in \boldsymbol{P}^{'}} \|p_i - p_i^{'}\|_2 + \frac{1}{|\boldsymbol{P}^{'}|} \sum_{p_i^{'} \in \boldsymbol{P}^{'}} \min_{p_i \in \boldsymbol{P}} \|p_i - p_i^{'}\|_2. \tag{10}$$

Altogether, the network is trained end-to-end by minimizing the following joint loss function

$$L = L_{local} + \gamma L_{global}, \tag{11}$$

where γ ($\gamma = 1$) is the proportion of two part errors.

4 EXPERIMENTS

In this section, we first investigate how some key parameters affect the performance of L2G-AE in the shape classification task on ModelNet10 [36]. Then, an ablation study is done to show the effectiveness of each module in L2G-AE. Finally, we further evaluate the performances of L2G-AE in multiple applications including 3D shape classification, 3D shape retrieval and point cloud upsampling.

4.1 Network Configuration

In L2G-AE, we first sample $M = 256$ points as the centroids of local regions by FPS. Then, around each centroid, a kNN searching algorithm selects $T = 4$ scale areas with $[K_1 = 16, K_2 = 32, K_3 = 64, K_4 = 128]$ points inside each area. In the multi-level feature propagation process, we initialize the feature dimension $C = M/8 = 32$ and $D = 256$. The encoder learns a 1024-dimension global feature for the input point cloud through hierarchical feature extraction. Similarly, the decoder hierarchically reconstructs local scales and global point cloud. In the RNN decoding layer, we adopt LSTM as the default RNN cell with hidden state dimension $h = D = 256$. In the experiment, we train our network on a NVIDIA GTX 1080Ti GPU using ADAM optimizer with the initial learning rate of 0.0001 and batch size of 8. The learning rate is decreased by 0.3 for every 20 epochs.

4.2 Parameters

All experiments on parameter comparison are evaluated under ModelNet10. ModelNet10 contains 4899 CAD models from 10 categories and is split into 3991 for training and 908 for testing. For each model, we adopt 1024 points which are uniformly sampled from mesh faces and are normalized into a unit ball before being fed into the network. During the training process, the loss function keeps decreasing and stabilizes around the 180th epoch. To acquire the accuracies on ModelNet10, we train a linear SVM from the global features obtained by the auto-encoder. Specifically, the OneVsRest strategy is adopted with the linearSVM function as the kernel.

We first explore the number of sampled points M which determines the distribution of local regions inside point clouds. In the experiment, we keep the network settings as depicted in the network configuration and vary the number of sampled points M from 128 to 320. The results are shown in Table 1, where the instance accuracies on the benchmark of ModelNet10 have a tendency to rise

Table 1: The effects of the number of sampled points M under ModelNet10.

M	128	192	256	320
Acc (%)	93.83	94.38	**95.37**	93.94

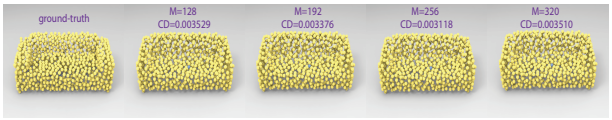

Figure 5: The reconstructed results with different sampled points, where the CD represents the Chamfer distance between ground-truth and the reconstructed point cloud.

first and then fall. This comparison implies that L2G-AE can effectively extract the contextual information in point clouds by multi-level feature propagation and $M = 256$ is an optimal choice which can well cover input point clouds without excessive redundant. To learn the reconstructed results intuitively, Figure 5 shows the reconstructed point clouds with different sampled points. According to Chamfer distances, L2G-AE can also reconstruct the input point cloud with the varying of sampled points.

With keeping the sampled points $M = 384$, we investigate the key parameter dimension C inside the self-attention modules. To unify the parameter in self-attention module, we keep the same dimension C in different semantic levels. We change the default $C = 32$ to 16 and 64, respectively. In Table 2, L2G-AE achieves the best performance when the feature dimension C is 32. Finally,

Table 2: The effects of the feature dimension C of the self-attention module under ModelNet10.

M	16	32	64
Acc (%)	93.94	**95.37**	94.16

we show the effects of feature dimension of local areas D and the global feature D_{global}. The dimension is varied as shown in Table 3 and Table 4. Neither the biggest nor the smallest, L2G-AE gets better performances when D, D_{global} are set to 256 and 1024 respectively. There is a trade-off between the network complexity and the expressive ability of our L2G-AE.

Table 3: The effects of the local feature dimension D on ModelNet10.

D	128	256	512
Acc (%)	93.72	**95.37**	93.28

4.3 Ablation Study

To quantitatively evaluate the effect of the self-attention module, we show the performances of L2G-AE under four settings: with point level self-attention module only (PL), with area level self-attention module only (AL), with region level self-attention module only (RL), remove all self-attention modules (NSA) and with

Table 4: The effects of the global feature dimension D_{global} under ModelNet10.

D_{global}	512	1024	2048
Acc (%)	94.16	**95.37**	93.94

Figure 6: The reconstruction results of L2G-AE with only the local loss and only the global loss.

all self-attention modules (ASA). As shown in Table 5, the self-attention module is effective in learning highly discriminative representations of point clouds by capturing the correlation among feature vectors. The results with only one self-attention module outperform the results without any self-attention module. And we achieve the best performance when three self-attention modules work together. The performance of self-attentions is affected by the discriminative ability of features. At the area level, the features of areas in the same region are similar, since there are only four areas, which makes the self-attention at area level contribute the least among all three self-attentions. In contrast, at the point level and the region level, the features of points or regions change a lot, so these self-attentions contribute more. From our observation, the results of PL and RL are coincidentally equal in the experiments.

Table 5: The effects of the self-attention module on ModelNet10.

Metric	PL	AL	RL	NSA	ASA
Acc (%)	94.16	94.05	94.16	93.72	**95.37**

After exploring the self-attention module, we also discuss the contributions of the two loss functions L_{local} and L_{global}. In Table 6, the results with local loss only (Local), global loss only (Global) and two losses together (Local + Global) are listed. The local loss function is very important in capturing local patterns of point clouds. And the two loss functions together can further enhance the classification performances of our neural network. In addition, Figure 6 shows the reconstruction results of our L2G-AE with only local loss and only global loss, respectively. From the results of the reconstructed point clouds, L2G-AE can reconstruct the input point cloud with only part of the joint loss function. In particular, the local reconstructed result in Figure 6 is a dense point cloud.

Table 6: The effects of the two loss functions L_{local} and L_{global} on ModelNet10.

Metric	Local	Global	Local+Global
Acc (%)	94.71	92.84	**95.37**

Table 7: The comparison of classification accuracy (%) under ModelNet10 and ModelNet40.

Methods	Supervised	MN40	MN10
PointNet	Yes	89.20	-
PointNet++	Yes	90.70	-
ShapeContextNet	Yes	90.00	-
Kd-Net	Yes	91.80	94.00
KC-Net	Yes	91.00	94.4
PointCNN	Yes	92.20	-
DGCNN	Yes	92.20	-
SO-Net	Yes	90.90	94.1
Point2Sequence	Yes	92.60	95.30
MAP-VAE	No	90.15	94.82
LGAN	No	85.70	95.30
LGAN(MN40)	No	87.27	92.18
FoldingNet	No	88.40	94.40
FoldingNet(MN40)	No	84.36	91.85
Our	No	**90.64**	**95.37**

4.4 Classification

In this subsection, we evaluate the performance of L2G-AE under ModelNet10 and ModelNet40 benchmarks, where ModelNet40 contains 12, 311 CAD models which is split into 9, 843 for training and 2, 468 for testing. Table 7 compares L2G-AE with state-of-the-art methods in the shape classification task on ModelNet10 and ModelNet40. The compared methods include PointNet [28], PointNet++ [29], ShapeContextNet [37], KD-Net [21], KC-Net [31], PointCNN [24], DGCNN [35], SO-Net [23], Point2Sequence [26], MAP-VAE [16], LGAN [1] and FoldingNet [40].

L2G-AE significantly outperforms all the unsupervised competitors under ModelNet10 and ModelNet40, respectively. In particular, L2G-AE achieves accuracy 95.37% which is even higher than other methods of supervision under ModelNet10. Although the results of LGAN [1] and FoldingNet [40] also show good performances under ModelNet10 and ModelNet40. This is because these methods are trained under a version of ShapeNet55 that contains more than 57,000 3D shapes. However, this version of ShapeNet55 dataset is not avaiable for public download from the official website. Therefore, we train all these methods under ModelNet40 for the fair comparison.

Table 8: The comparison of retrieval in terms of under ModelNet10.

Methods	LGAN	FoldingNet	Our
Acc (%)	49.94	53.42	**67.81**

4.5 Retrieval

L2G-AE is further evaluated in the shape retrieval task under ModelNet10 and compared with some other unsupervised methods of learning on point clouds. The compared results include two state-of-the-art unsupervised methods for point clouds, i.e., LGAN [1] and FoldingNet [40]. The target of shape retrieval is to obtain the

Table 9: The quantitative comparison of $16\times$ upsampling from 625 points under ModelNet10.

10^{-3}	bathtub	bed	chair	desk	dresser	monitor	n.stand	sofa	table	toilet
PU	**1.01**	**1.12**	**0.82**	**1.22**	1.55	**1.19**	**1.77**	**1.13**	**0.69**	**1.39**
EC	1.43	1.81	1.80	1.30	1.43	2.04	1.88	1.79	1.00	1.72
Our	1.74	1.46	1.58	2.08	**1.40**	1.61	1.86	1.67	1.86	2.10

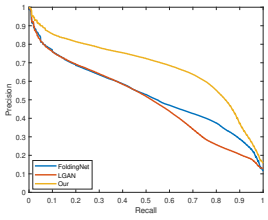

Figure 7: The comparison of PR curves for retrieval under ModelNet10.

relevant information of a inquiry from a collection. In these experiments, the 3D shapes in the test set are used as quires to retrieve the rest shapes in the same set, and mean Average Precision (mAP) is used as a metric.

As shown in Table 8, our results outperform all the compared results under ModelNet10. It shows that L2G-AE can be effect in improving the performance of unsupervised shape retrieval on point clouds. Their PR curves under ModelNet10 are also compared in Figure 7 which intuitively shows the performances of these three methods.

4.6 Unsupervised Upsampling for Point Clouds

Benefit from the design of local to global reconstruction, it is competent for our L2G-AE to be applied in the unsupervised point cloud upsampling application. In the local reconstruction, a dense point cloud is obtained by reconstructing multiple local scales with overlapping. Therefore, it is convenient to produce the upsampling results by downsampling from the dense local reconstructed results using some unsupervised methods, such as random sampling or farthest point sampling. As far as we know, L2G-AE is the first method which performs point cloud upsampling with deep neural networks in an unsupervised manner. To evaluate the performance of L2G-AE, We compare our method on relatively sparse (625 points) inputs with state-of-the-art supervised point cloud upsampling methods, including PU-Net [44] and EC-Net [43]. The target of upsampling is to generate a dense point clouds with 10000 points. For PU-Net and EC-Net, the $16\times$ results (10000 points) are obtained from inputs (625 points) in a supervised manner. Differently, L2G-AE first obtains the local reconstruction results and then downsamples them to 10000 points.

As shown in Table 9, mean Chamfer Distance (mCD) is used as a metric for quantitative comparison with PU-Net (PU) and EC-Net (EC) under ModelNet10. Although the results of PU-Net and EC-Net are better than "Our" in some classes under ModelNet10, the most likely reason is that the ground-truth is not visible to L2G-AE in the training. In addition, the input point cloud with 625 points

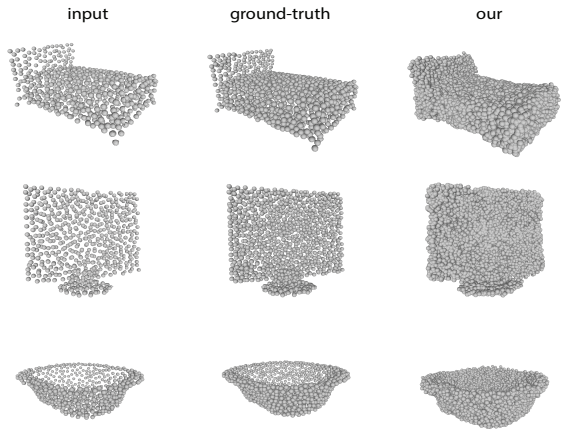

Figure 8: Some upsampled results of L2G-AE.

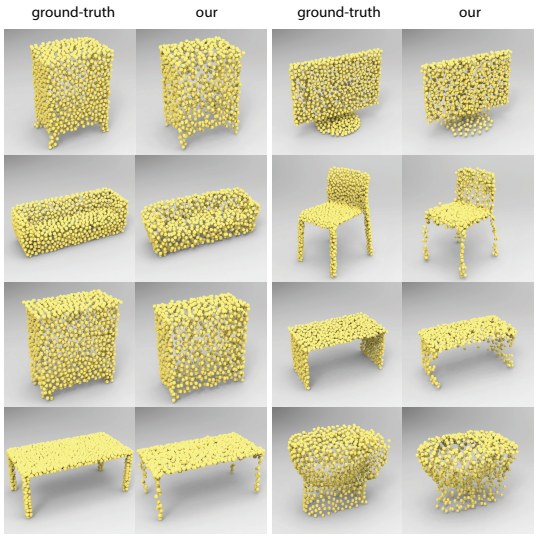

Figure 9: Some reconstructed examples of L2G-AE.

contains very limited information. Figure 8 shows some upsamled results of our L2G-AE.

4.7 Visualization

In this section,we will show some important visualization results of L2G-AE. Firstly, some reconstructed point clouds by L2G-AE are listed with the ground-truths as shown in Figure 9. From the results, the reconstructed point clouds of L2G-AE are consistent with the ground-truths.

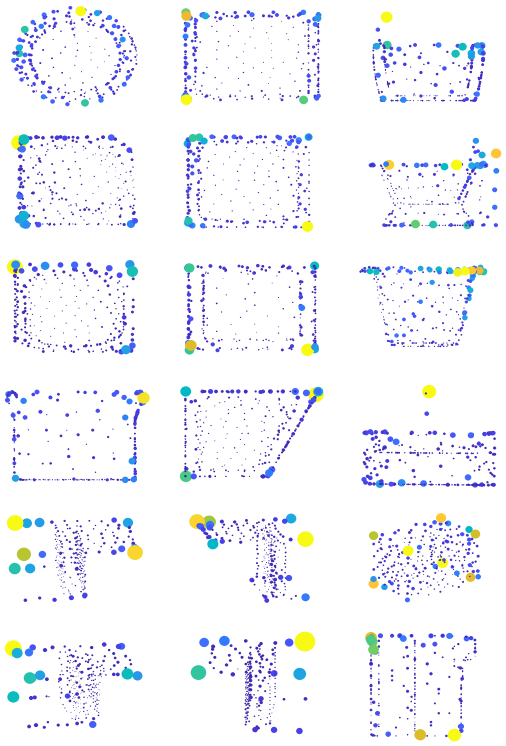

Figure 10: Some examples of the attention in the region level, where each subfigure represents a 3D object.

Then, some visualizations of the attention map inside self-attention modules are engaged to show the effect of attentions in the hierarchical feature abstraction. There are three self-attention modules in the encoder, and we first visualize the attention map inside the local region level. For intuitively understanding, we directly attach the attention values to the centroids of local regions and then show these centroids. By summing attention map by column in the region level, the attention value of each centroid is caculated. For example, a 256×256 attention map is translated to a 256-dimension attention vector, when the number of sampled centroids is 256. Then, both the size and the color of centroids are associated with the attention values. Therefore, the centroids with lighter colors and larger sizes indicate larger attention values. As depicted in Figure 10, we show some examples of the region level attention. Figure 10 shows that the self-attention in the region level tends to on special local regions at conspicuous locations such as edges, corners or protruding parts.

Similarly, we also show some examples of the scale level attention in Figure 11 and the point level attention in Figure 12. In Figure 11, each image shows the 4 scale attention values around 256 sampled centroids of a point cloud. And the color indicates the value of attention, where large attention value corresponds to a bright color such as yellow. The results indicate that the network tends to focus on the 4^{th} scale which contains more information of local structures. In Figure 12, each row represents the 4 scale areas around a centroid. In different scale areas, the network concern on

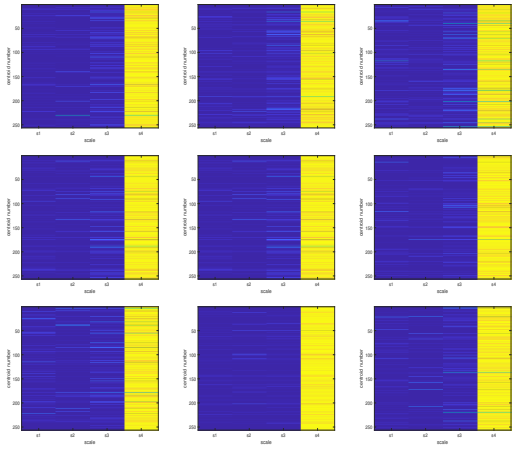

Figure 11: Some examples of the attention in the scale level. The abscissa represents the 4 scales $[s_1, s_2, s_3, s_4]$ around each centroid in a point cloud and the ordinate indicates the index of 256 centroids, where each subfigure represents a 3D object.

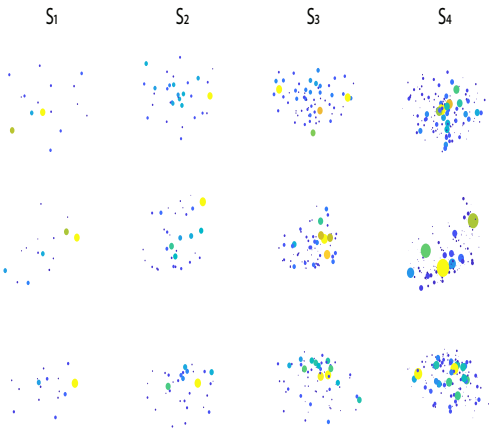

Figure 12: Some examples of the attention in the point level, where the four subfigures in each row represent the four scales of a local region.

different points inside the areas to capture the local patterns in the local region.

5 CONCLUSIONS

In this paper, we propose a novel local to global Auto-encoder framework for point cloud understanding in the shape classification, retrieval and point cloud upsampling tasks. In the encoder, a self-attention mechanism is employed to explore the correlation among features in the same level. In contrast, an interpolation layer and RNN decoding layer successfully reconstruct local scales and global point clouds hierarchically. Experimental results show that our method achieves competitive performances with state-of-the-art methods.

6 ACKNOWLEDGMENTS

Yu-Shen Liu is the corresponding author. This work was supported by National Key R&D Program of China (2018YFB0505400), the National Natural Science Foundation of China (61472202), and National Science Foundation grant (1813583). We thank all anonymous reviewers for their constructive comments.

REFERENCES

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2017. Learning representations and generative models for 3D point clouds. In *arXiv preprint arXiv:1707.02392*.

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. 2001. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*. 831–837.

[3] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 2018. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. (2018).

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

[5] Aleksey Golovinskiy, Vladimir G Kim, and Thomas Funkhouser. 2009. Shape-based recognition of 3D point clouds in urban environments. In *ICCV*. 2154–2161.

[6] Zhizhong Han, Xinhai Liu, Yu-Shen Liu, and Matthias Zwicker. 2019. Parts4Feature: Learning 3D Global Features from Generally Semantic Parts in Multiple Views. *arXiv preprint arXiv:1905.07506* (2019).

[7] Zhizhong Han, Zhenbao Liu, Junwei Han, ChiMan Vong, Shuhui Bu, and C.L.P. Chen. 2019. Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy. *IEEE Transactions on Cybernetics* 49, 2 (2019), 481–494.

[8] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and C.L.Philip Chen. 2017. Mesh Convolutional Restricted Boltzmann Machines for Unsupervised Learning of Features With Structure Preservation on 3D Meshes. *IEEE Transactions on Neural Network and Learning Systems* 28, 10 (2017), 2268 – 2281.

[9] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and Xuelong Li. 2016. Unsupervised 3D Local Feature Learning by Circle Convolutional Restricted Boltzmann Machine. *IEEE Transactions on Image Processing* 25, 11 (2016), 5331–5344.

[10] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. 2017. BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation. *IEEE Transactions on Image Processing* 26, 8 (2017), 3707–3720.

[11] Zhizhong Han, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Shuhui Bu, Junwei Han, and CL Philip Chen. 2018. Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. *IEEE Transactions on Image Processing* 27, 6 (2018), 3049–3063.

[12] Zhizhong Han, Honglei Lu, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liua, Matthias Zwicker, Junwei Han, and CL Philip Chen. 2019. 3D2SeqViews: Aggregating Sequential Views for 3D Global Feature Learning by CNN with Hierarchical Attention Aggregation. *IEEE Transactions on Image Processing* (2019).

[13] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. 2018. View inter-prediction gan: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. *arXiv preprint arXiv:1811.02744* (2018).

[14] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. 2018. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing* 28, 2 (2018), 658–672.

[15] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 2018. Y^ 2Seq2Seq: Cross-Modal Representation Learning for 3D Shape and Text by Joint Reconstruction and Prediction of View and Word Sequences. *arXiv preprint arXiv:1811.02745* (2018).

[16] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 2019. Multi-Angle Point Cloud-VAE: Unsupervised Feature Learning for 3D Point Clouds from Multiple Angles by Joint Self-Reconstruction and Half-to-Half Prediction. *ICCV* (2019).

[17] Zhizhong Han, Xiyang Wang, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, and CL Chen. 2019. 3DViewGraph: Learning Global Features for 3D Shapes from A Graph of Unordered Views with Attention. *arXiv preprint arXiv:1905.07503* (2019).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *CVPR*. 4700–4708.

[21] Roman Klokov and Victor Lempitsky. 2017. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *ICCV*. 863–872.

[22] Hema S Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. 2011. Semantic labeling of 3D point clouds for indoor scenes. In *NIPS*. 244–252.

[23] Jiaxin Li, Ben M Chen, and Gim Hee Lee. 2018. SO-Net: Self-Organizing Network for Point Cloud Analysis. In *CVPR*. 9397–9406.

[24] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. PointCNN: Convolution on X-Transformed points. In *NIPS*.

[25] Yangyan Li, Soeren Pirk, Hao Su, Charles R Qi, and Leonidas J Guibas. 2016. FPNN: Field probing neural networks for 3D data. In *NIPS*. 307–315.

[26] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. 2018. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. *arXiv preprint arXiv:1811.02565* (2018).

[27] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. 2017. Frustum pointNets for 3D object detection from RGB-D data. In *CVPR*.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2016. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*.

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*. 5099–5108.

[30] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, Vol. 3.

[31] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. 2018. Mining point cloud local structures by kernel correlation and graph pooling. In *CVPR*, Vol. 4.

[32] M Simon, S Milz, K Amende, and HM Gross. 2018. Complex-YOLO: Real-time 3D object detection on point clouds. *arXiv preprint arXiv:1803.06199* (2018).

[33] David Stutz and Andreas Geiger. 2018. Learning 3D shape completion from laser scan data with weak supervision. In *CVPR*.

[34] Yuan Wang, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. 2018. PointSeg: Real-Time semantic segmentation based on 3D LiDAR point cloud. In *arXiv preprint arXiv:1807.06288*.

[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2018. Dynamic graph CNN for learning on point clouds. In *arXiv preprint arXiv:1801.07829*.

[36] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*. 1912–1920.

[37] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. 2018. Attentional ShapeContextNet for point cloud recognition. In *CVPR*. 4606–4615.

[38] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. 2018. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In *ECCV*.

[39] Bin Yang, Wenjie Luo, and Raquel Urtasun. 2018. PIXOR: Real-time 3D object detection from point clouds. In *CVPR*.

[40] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR*.

[41] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. 2018. Patch-base progressive 3D point set upsampling. In *arXiv preprint arXiv:1811.11286*.

[42] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. 2018. PVNet: A joint convolutional network of point cloud and multi-view for 3D shape recognition. In *ACM Multimedia Conference*.

[43] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2018. EC-Net: an Edge-aware point set consolidation network. In *ECCV*.

[44] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2018. PU-Net: Point cloud upsampling network. In *CVPR*.

[45] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. In *NIPS*.

[46] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3D point-capsule networks. In *CVPR*.

[47] Yin Zhou and Oncel Tuzel. 2017. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *CVPR*.