Multi-Angle Point Cloud-VAE: Unsupervised Feature Learning for 3D Point Clouds from Multiple Angles by Joint Self-Reconstruction and Half-to-Half Prediction

Zhizhong Han[1,3], Xiyang Wang[1,2], Yu-Shen Liu[1,2]*, Matthias Zwicker[3]
[1]School of Software, Tsinghua University, Beijing, China
[2]Beijing National Research Center for Information Science and Technology (BNRist)
[3]Department of Computer Science, University of Maryland, College Park, USA
h312h@umd.edu, wangxiya16@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn, zwicker@cs.umd.edu

Abstract

Unsupervised feature learning for point clouds has been vital for large-scale point cloud understanding. Recent deep learning based methods depend on learning global geometry from self-reconstruction. However, these methods are still suffering from ineffective learning of local geometry, which significantly limits the discriminability of learned features. To resolve this issue, we propose MAP-VAE to enable the learning of global and local geometry by jointly leveraging global and local self-supervision. To enable effective local self-supervision, we introduce multi-angle analysis for point clouds. In a multi-angle scenario, we first split a point cloud into a front half and a back half from each angle, and then, train MAP-VAE to learn to predict a back half sequence from the corresponding front half sequence. MAP-VAE performs this half-to-half prediction using RNN to simultaneously learn each local geometry and the spatial relationship among them. In addition, MAP-VAE also learns global geometry via self-reconstruction, where we employ a variational constraint to facilitate novel shape generation. The outperforming results in four shape analysis tasks show that MAP-VAE can learn more discriminative global or local features than the state-of-the-art methods.

1. Introduction

Point clouds have become a popular 3D representation in machine vision, autonomous driving, and augmented reality, because they are easy to acquire and manipulate. Therefore, point cloud analysis has emerged as a crucial problem in the area of 3D shape understanding. With the help of extensive supervised information, recent deep learning based feature learning techniques have achieved unprecedented results in classification, detection and segmentation [27, 31, 22, 35, 23, 39]. However, supervised learning requires intense manual labeling effort to obtain supervised information. Therefore, unsupervised feature learning is an attractive alternative and a promising research challenge.

Several studies have tried to adress this challenge [1, 21, 4, 34, 33, 42, 32]. To learn the structure of a point cloud without additional supervision, these generative models are trained by self-supervision, such as self-reconstruction [1, 4, 6, 42, 33, 32] or distribution approximation [1, 21, 34], which is implemented by auto-encoder or generative adversarial networks [8] respectively. To capture finer global structure, some methods [34, 33, 42, 32] first learn local structure information in point cloud patches based on which the global point cloud is then reconstructed. Because of lacking effective and semantic local structure supervision, however, error may accumulate in the local structure learning process, which limits the network's ability in 3D point cloud understanding.

To resolve this issue, we propose a novel deep learning model for unsupervised point cloud feature learning by simultaneously employing effective local and global self-supervision. We introduce multi-angle analysis for point clouds to mine effective local self-supervision, and combine it with global self-supervision under a variational constraint. Hence we call our model Multi-Angle Point Cloud Variational Auto-Encoder (MAP-VAE). Specifically, to employ local self-supervision, MAP-VAE first splits a point cloud into a front half and a back half under each of several incrementally varying angles. Then, MAP-VAE performs half-to-half prediction to infer a sequence of several back halves from the corresponding sequence of the complementary front halves. Half-to-half prediction aims to capture the geometric and structural information of local regions on the point cloud through varying angles. More-

*Corresponding Author. This work was supported by National Key R&D Program of China (2018YFB0505400) and NSF (award 1813583).

over, by leveraging global self-supervision, MAP-VAE conducts self-reconstruction in company with each half-to-half prediction to capture the geometric and structural information of the whole point cloud. Self-reconstruction is started from a variational feature space, which enables MAP-VAE to generate new shapes by capturing the distribution information over training point clouds in the feature space. In summary, our contributions are as follows:

i) We propose MAP-VAE to perform unsupervised feature learning for point clouds. It can jointly leverage effective local and global self-supervision to learn fine-grained geometry and structure of point clouds.

ii) We introduce multi-angle analysis for point cloud understanding, which provides semantic local self-supervision to learn local geometry and structure.

iii) We provide a novel way to consistently split point clouds into semantic regions according to view angles, which enables the exploration of the fine-grained discriminative information of point cloud regions.

2. Related work

Deep learning models have led to significant progress in feature learning for 3D shapes [12, 11, 14, 13, 17, 18, 9, 19, 15, 10]. Here, we focus on reviewing studies on point clouds. For supervised methods, supervised information, such as shape class labels or segmentation labels, are required to train deep learning models in the feature learning process. In contrast, unsupervised methods are designed to mine self-supervision information from point clouds for training, which eliminates the need for supervised information that can be tedious to obtain. We briefly review the state-of-the-art methods in these two categories as follows.

Supervised feature learning. As a pioneering work, PointNet [26] was proposed to directly learn features from point clouds by deep learning models. However, PointNet is limited in capturing contextual information among points. To resolve this issue, various techniques were proposed to establish graph in a local region to capture the relationship among points in the region [31, 22, 35, 23, 39]. Furthermore, multi-scale analysis [27] was introduced to extract more semantic features from the local region by separating points into scales or bins, and then, aggregating these features by concatenation [38] or RNN [25]. These methods require supervised information in the feature learning process, which is different from unsupervised approach in MAP-VAE.

Unsupervised feature learning. An intuitive approach to mine self-supervised information is to perform self-reconstruction which first encodes a point cloud into a feature and then decodes the feature back to a point cloud. Such global self-supervision is usually implemented by an autoencoder network [1, 4, 42, 33, 32]. With the help of adversarial training, a different kind of global self-supervision is employed to train the network to generate plausible point clouds by learning a mapping from a known distribution to the unknown distribution that the point clouds are sampled from [1, 21, 34]. For finer global structure information, some methods take a step further to jointly employ local structure information captured in local regions [34, 33, 42, 32]. These methods first learn local structure information in point cloud patches by clustering [42], conditional point distribution [33], graph convolution [34], or fully connected layers [32], based on which the whole point cloud is then reconstructed. However, because of lacking effective and semantic local structure supervision, this process is prone to error accumulation in the local structure learning process, which limits the network's ability in point cloud understanding. To resolve this issue, MAP-VAE introduces multi-angle analysis for point clouds, which provides effective and semantic local self-supervision. MAP-VAE can also simultaneously employ local and global self-supervision, which further differentiates it from others.

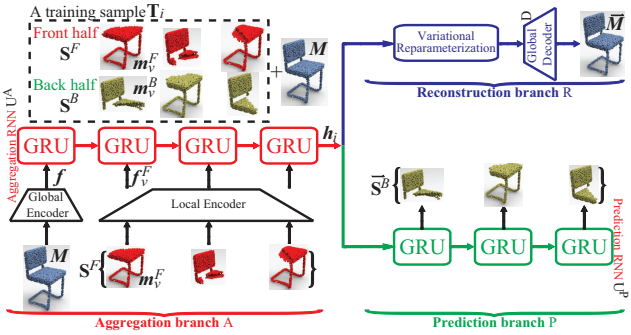

Figure 1. The framework of MAP-VAE.

3. Overview

To jointly leverage local and global self-supervision to learn features from point clouds, MAP-VAE simultaneously conducts half-to-half prediction and self-reconstruction by three branches, i.e., which we call aggregation branch A, reconstruction branch R, and prediction branch P, as illustrated in Fig. 1. Specifically, branch A and branch P together perform the half-to-half prediction while branch A and branch R together perform the self-reconstruction.

A training sample $\mathbf{T}_i$ provided to MAP-VAE to learn is formed by a front half sequence $\mathbf{S}^F$, a back half sequence $\mathbf{S}^B$, and an original point cloud $\boldsymbol{M}$. The corresponding elements in sequences $\mathbf{S}^F$ and $\mathbf{S}^B$ are a front half $\boldsymbol{m}_v^F$ (in red) and its complementary back half $\boldsymbol{m}_v^B$ (in green) which are obtained by splitting the original point cloud $\boldsymbol{M}$ (in blue) from a specific angle v.

The aggregation branch A encodes the geometry of local

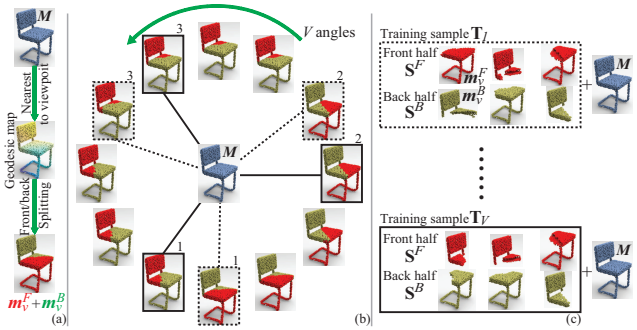

Figure 2. (a) Geodesic splitting $\boldsymbol{M}$ into a front half $\boldsymbol{m}_v^F$ (in red) and a back half $\boldsymbol{m}_v^B$ (in green) from the v-th angle. (b) $\boldsymbol{M}$ is further split from all V angles located around $\boldsymbol{M}$ in clockwise order, where subset with $W = 3$ out of V angles (indicated by dotted or solid line) are selected to establish a half-to-half sequence pair. (c) The training sample $\mathbf{T}_i$ from each one of V angles.

point clouds and their spatial relationship by aggregating all front halves in sequence $\mathbf{S}^F$ in order. It first extracts the low-level feature $\boldsymbol{f}$ of the original point cloud $\boldsymbol{M}$ and the low-level feature $\boldsymbol{f}_v^F$ of each front half $\boldsymbol{m}_v^F$ by a global encoder and a local encoder, respectively. Then, it learns the angle-specific feature $\boldsymbol{h}_i$ of $\boldsymbol{M}$ by aggregating all low-level features $\boldsymbol{f}_v^F$ using an aggregation RNN $\mathrm{U^A}$.

The reconstruction branch R performs self-reconstruction by decoding the learned angle-specific feature $\boldsymbol{h}_i$ into a point cloud $\overline{\boldsymbol{M}}$. This reconstruction is conducted by a global decoder D which tries to generate $\overline{\boldsymbol{M}}$ as similar as possible to $\boldsymbol{M}$. In addition, R employs a variational constraint to facilitate novel shape generation.

At the same time, the prediction branch P performs half-to-half prediction by decoding the learned angle-specific feature $\boldsymbol{h}_i$ into a back half sequence $\overline{\mathbf{S}^B}$ which is paired with the corresponding front half sequence $\mathbf{S}^F$. This prediction is conducted by a prediction RNN $\mathrm{U^P}$ which tries to predict the sequence $\overline{\mathbf{S}^B}$ as similar as possible to $\mathbf{S}^B$.

4. Multi-angle splitting

Multi-angle splitting. A key idea in MAP-VAE is a novel multi-angle analysis for point coulds to mine effective local self-supervision. Intuitively, observing a point cloud from different angles, explicitly presents the correspondences and relationships among different shape regions, given as the correspondence between the front and back halves of the shape in each view. Our multi-angle analysis provides multiple regions (front halves) of the point cloud as input, from which the corresponding missing regions (back halves) need to be inferred. This encourages MAP-VAE to learn a detailed shape representation that facilitates high quality classification, segmentation, shape synthesis, and point cloud completion.

We achieve this by splitting a point cloud into a front

Figure 3. The comparison of front halves (red parts in (a) and (c)) split by Euclidean distance and geodesic distance (map in (b)).

and a back half from different angles, where the front half is the half nearer to the viewpoint than the back half. This enables MAP-VAE to observe different semantic parts of a point cloud, and it also preserves the spatial relationship among the parts by incrementally varying angles.

For a point cloud $\boldsymbol{M}$, we split $\boldsymbol{M}$ from V different angles into front halves (in red) and their complementary back halves (in green), as shown in Fig. 2 (b), where the viewpoints are located around $\boldsymbol{M}$ on a circle. From the v-th viewpoint, $\boldsymbol{M}$ is split into a front half $\boldsymbol{m}_v^F$ and a back half $\boldsymbol{m}_v^B$ in Fig. 2 (a), where $\boldsymbol{m}_v^F$ is formed by the N nearest points (in red) of $\boldsymbol{M}$ to the viewpoint while $\boldsymbol{m}_v^B$ is formed by the N farthest points (in green) to the viewpoint.

Geodesic splitting. A naive way of finding the N nearest points to define a front half $\boldsymbol{m}_v^F$ is to sort all points on $\boldsymbol{M}$ by the Euclidean distance between each point and the viewpoint. However, on some point clouds, this method may not produce semantic front halves, since the regions in a front half are not continuous, as demonstrated in Fig. 3 (a). It is important to encode semantic front halves, since this would help MAP-VAE to seamlessly understand the entire surface from a viewpoint under a multi-angle scenario.

To resolve this issue, we leverage the geodesic distance on the point cloud [2] to sort the points. Specifically, we first find a nearest point $\boldsymbol{u}$ to the v-th viewpoint on $\boldsymbol{M}$ by Euclidean distance. Then, we sort the rest of points on $\boldsymbol{M}$ in terms of their geodesic distances to the found nearest point $\boldsymbol{u}$, as shown by the geodesic map in Fig. 3 (b). Finally, $\boldsymbol{u}$ and its nearest $N-1$ points form the front half $\boldsymbol{m}_v^F$, while the farthest N points form the back half $\boldsymbol{m}_v^B$, as illustrated by the red part and green part in Fig. 3 (c), respectively.

Half-to-half sequence pairs. To leverage the correspondence between front half and back half and their spatial relationship under different angles, we establish a half-to-half sequence pair starting from each one of the V angles.

Along the circle direction of varying angles, as illustrated by the clockwise green arrow in Fig. 2 (b), we select front halves $\boldsymbol{m}_v^F$ and their complementary back halves $\boldsymbol{m}_v^B$ from W out of the V angles. The selected $\boldsymbol{m}_v^F$ form a front half sequence $\mathbf{S}^F$ while the complementary $\boldsymbol{m}_v^B$ form a back half sequence $\mathbf{S}^B$, where $\mathbf{S}^F = [\boldsymbol{m}_v^F | v \in [1, V], |v| = W]$, $\mathbf{S}^B = [\boldsymbol{m}_v^B | v \in [1, V], |v| = W]$ and each element in $\mathbf{S}^F$ corresponds to its complementary element in $\mathbf{S}^B$. Thus, a half-to-half sequence pair $(\mathbf{S}^F, \mathbf{S}^B)$ consists of $\mathbf{S}^F$ and $\mathbf{S}^B$.

To comprehensively observe the point cloud, we select W angles which uniformly cover the whole shape in each

half-to-half sequence pair $(\mathbf{S}^F, \mathbf{S}^B)$. As demonstrated by the dotted lines in Fig. 2 (b), we select $W = 3$ angles in order to form the first $(\mathbf{S}^F, \mathbf{S}^B)$, and then, along the green arrow, we form the last $(\mathbf{S}^F, \mathbf{S}^B)$ by angles indicated by the solid lines. Each $(\mathbf{S}^F, \mathbf{S}^B)$ forms a training sample $\mathbf{T}_i$ in company with the point cloud $\boldsymbol{M}$. Finally, we obtain all V training samples $\{\mathbf{T}_i | i \in [1, V]\}$ from $\boldsymbol{M}$ in Fig. 2 (c).

5. MAP-VAE

Aggregation branch A**.** For a training sample $\mathbf{T}_i$ containing a half-to-half sequence pair $(\mathbf{S}^F, \mathbf{S}^B)$ and the point cloud $\boldsymbol{M}$, aggregation branch A encodes the global geometry of $\boldsymbol{M}$, local geometry of each one of W $\mathbf{m}_v^F$ in $\mathbf{S}^F$, and the spatial relationship among $\mathbf{m}_v^F$. Aggregation branch A first extracts the geometry of each involved point cloud into a low-level feature by encoder, and then, aggregates all the low-level features with their spatiality by aggregation RNN U^{A}. Specifically, we extract the low-level feature $\boldsymbol{f}$ of $2N$ points on $\boldsymbol{M}$ by a global encoder, and the low-level feature $\boldsymbol{f}_v^F$ of N points on $\mathbf{m}_v^F$ by a local encoder. Both the global and local encoders employ the same architecture as the encoder in PointNet++ [27], the only difference is the input number of points. Subsequently, aggregation RNN U^{A} aggregates $\boldsymbol{f}$ and all $\boldsymbol{f}_v^F$ in $W + 1$ steps, where we employ GRU cell with 512 hidden state. Finally, we use the hidden state as the angle-specific feature $\boldsymbol{h}_i$ of $\boldsymbol{M}$ since the first front half in $\mathbf{S}^F$ is observed starting from the i-th angle.

Reconstruction branch R**.** By decoding the learned angle-specific feature $\boldsymbol{h}_i$, reconstruction branch R tries to generate a point cloud $\overline{\boldsymbol{M}}$ as similar as possible to the original point cloud $\boldsymbol{M}$ by a global decoder D. D is formed by 3 fully connected layers (1024-2048-6114) and 2 convolutional layers (with 256 and 3 1×1 kernels each), where batch normalization is used between every two layers. Here, we prefer Earth Movers distance (EMD) [29] to Chamfer Distance (CD) [5] to evaluate the distance between the reconstructed $\overline{\boldsymbol{M}}$ and the original $\boldsymbol{M}$, since EMD is more faithful than CD to the visual quality of point clouds[1]. The EMD distance between $\overline{\boldsymbol{M}}$ and $\boldsymbol{M}$ is regarded as the cost of reconstruction to optimize, as defined below, where ϕ is a bijection from a point x on $\boldsymbol{M}$ to its corresponding point $\phi(x)$ on $\overline{\boldsymbol{M}}$,

$$C_{\mathrm{D}} = \min_{\phi: \boldsymbol{M} \to \overline{\boldsymbol{M}}} \sum_{x \in \boldsymbol{M}} \|x - \phi(x)\|_2. \tag{1}$$

In addition, we employ a variational constraint [20] in reconstruction branch R to facilitate novel shape generation. This is implemented by a variational reparameterization process, as shown in Fig. 1. The variational reparameterization transforms the angle-specific feature $\boldsymbol{h}_i$ into another latent vector $\boldsymbol{z}$ that roughly follows a unit multi-dimensional Gaussian distribution. After training, branch R can generate a novel shape by sampling a latent vector from the unit Gaussian to the global decoder D.

Specifically, the variational reparameterization first employs fully connected layers to respectively estimate the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ for the distribution of $\boldsymbol{h}_i$. Then, a noise vector $\boldsymbol{\varepsilon}$ is sampled from a unit multi-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{1})$, as $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and $\boldsymbol{\varepsilon} \in \mathrm{R}^{1 \times Z}$. Finally, we scale the noise $\boldsymbol{\varepsilon}$ by $\boldsymbol{\sigma}$ and further shift it by $\boldsymbol{\mu}$, such that the latent vector $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\varepsilon} \odot \boldsymbol{\sigma}$. The variational reparameterization enables reconstruction branch R to push the distribution $q(\boldsymbol{z}|\boldsymbol{h}_i)$ to follow the unit multi-dimensional Gaussian distribution by minimizing the KL divergence between the distribution $q(\boldsymbol{z}|\boldsymbol{h}_i)$ and $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

Thus, the cost of reconstruction branch R is defined based on Eq. (1) below, where α is a balance parameter.

$$C_{\mathrm{R}} = C_{\mathrm{D}} + \alpha \times \mathrm{KL}(q(\boldsymbol{z}|\boldsymbol{h}_i) || \mathcal{N}(\mathbf{0}, \mathbf{1})). \tag{2}$$

Prediction branch P**.** Similar to reconstruction branch R, prediction branch P decodes the learned angle-specific feature $\boldsymbol{h}_i$ to predict the back half sequence $\mathbf{S}^B$ corresponding to $\mathbf{S}^F$. Branch P tries to predict a back half sequence $\overline{\mathbf{S}^B}$ as similar as possible to $\mathbf{S}^B$ by a prediction RNN U^{P}. At each of W steps, U^{P} predicts one back half $\overline{\mathbf{m}_v^B}$ in the same order of elements in $\mathbf{S}^B$. This enables U^{P} to learn the half-to-half correspondence and the spatial relationship among the halves of $\boldsymbol{M}$. To further push MAP-VAE to comprehensively understand the point cloud, U^{P} predicts the low-level feature $\overline{\boldsymbol{f}_v^B}$ of each one of W back half $\overline{\mathbf{m}_v^B}$ rather than the spatial point coordinates of $\overline{\mathbf{m}_v^B}$, which is complementary to reconstruction branch R. The ground truth low-level feature $\boldsymbol{f}_v^B$ of $\mathbf{m}_v^B$ is also extracted by the local encoder in branch A. Thus, the cost of branch P is defined as follows,

$$C_{\mathrm{P}} = \frac{1}{W} \times \sum_{v \in [1,V], |v| = W} \|\overline{\boldsymbol{f}_v^B} - \boldsymbol{f}_v^B\|_2^2. \tag{3}$$

Objective function. For a sample $\mathbf{T}_i$, MAP-VAE is trained by minimizing all the aforementioned costs of each branch, as defined below, where β is a balance parameter.

$$\min \; C_{\mathrm{R}} + \beta \times C_{\mathrm{P}}. \tag{4}$$

After training, MAP-VAE represents the point cloud $\boldsymbol{M}$ as a global feature $\boldsymbol{H}$ by aggregating the angle-specific feature $\boldsymbol{h}_i$ learned from each sample $\mathbf{T}_i$ of $\boldsymbol{M}$ using max pooling, such that $\boldsymbol{H} = \mathrm{Pool}_{i \in [1,V]}\{\boldsymbol{h}_i\}$.

6. Experimental results and analysis

In this section, we first explore how the parameters involved in MAP-VAE affect the discriminability of learned global features in shape classification. Then, MAP-VAE is evaluated in shape classification, segmentation, novel shape generation, and point cloud completion by comparing with state-of-the-art methods.

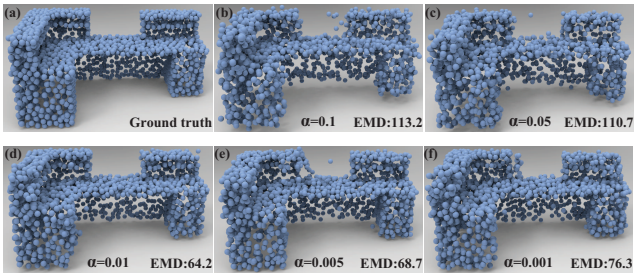

Figure 4. The point clouds are reconstructed in (b)-(f) under different α compared in Table. 2.

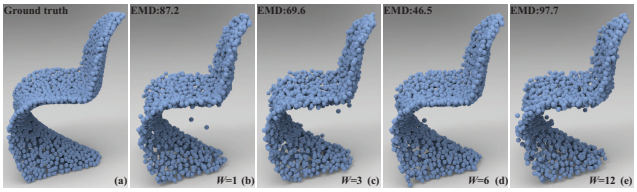

Figure 5. The original point clouds in (a) are reconstructed in (b)-(e) under different W compared in Table. 3.

Training. We pre-train the global and local encoders in MAP-VAE respectively under the dataset involved in experiments in a self-reconstruction task, where the decoder in PointNet++ [27] for segmentation is modified to work with our encoders to produce three dimensional point coordinates in the last layer. After each PointNet++-based autoencoder is trained, the pre-trained global and local encoders are fixed for more efficient training of MAP-VAE.

In all experiments, we choose a more challenging way to train MAP-VAE by all point clouds in multiple shape classes of a benchmark rather than a single shape class, where each point cloud has 2048 points and each half has $N = 1024$ points. In shape classification experiments, we train a linear SVM to evaluate the raw discriminability of the learned global feature $\boldsymbol{H}$.

Initially, we employ $V = 12$ angles to analyze a point cloud and form a training sample by $W = 6$ angles uniformly covering the point cloud. We set balance parameter $\alpha = 0.01$ and $\beta = 1000$ to make each cost in the same order of magnitude. We use a $Z = 128$ dimensional unit Gaussian for the variational constraint.

Parameter setting. All experiments on parameter effect exploration are conducted under ModelNet10 [37].

We first evaluate how β affects MAP-VAE by comparing the results of different β candidates including $\{10, 100, 1000, 10000\}$. As shown in Table 1, the results get better with increasing β until $\beta = 1000$ and degenerate when β is too big. This observation demonstrates a proper range of β. We use $\beta = 1000$ in the following experiments.

Then, we evaluate how α affects MAP-VAE by comparing the results of different α candidates including $\{0.1, 0.05, 0.01, 0.005, 0.001\}$. As shown in Table 2, the

Table 1. The effect of β, $\alpha = 0.01$, $W = 6$, $Z = 128$.

β	10	100	1000	10000
ACC%	93.72	93.94	**94.82**	93.72

Table 2. The effect of α, $\beta = 1000$, $W = 6$, $Z = 128$.

α	0.1	0.05	0.01	0.005	0.001
ACC%	92.62	92.84	**94.82**	93.39	93.17

Table 3. The effect of W, $\alpha = 0.01$, $\beta = 1000$, $Z = 128$.

W	1	3	6	12	S-6	R-6
ACC%	92.95	93.39	**94.82**	93.17	93.39	92.95

Table 4. The effect of Z, $\alpha = 0.01$, $\beta = 1000$, $W = 6$.

Z	32	64	128	256
ACC%	93.28	94.16	**94.82**	93.94

results get better with decreasing α until $\alpha = 0.01$ and degenerate when α is too small. This observation demonstrates how enforcing a unit Gaussian distribution on the latent vector $\boldsymbol{z}$ too loosely or strictly affects the discriminability of learned global features. We also visualize the point clouds reconstructed by branch R under different α, as demonstrated in Fig. 4. We find α affects the reconstructed point clouds in a similar way to how it affects the discriminability of learned global features. In the following experiments, we set α to 0.01.

Subsequently, we explore how the number of angles W of in a training sample affects the performance of MAP-VAE, as shown in Table. 3, where several candidate W including $\{1, 3, 6, 12\}$ are employed. We find $W = 6$ achieves the best result, where fewer angles provide less local information while more angles increase redundancy. We also observe a similar phenomenon in the reconstructed point clouds shown in Fig.5. In addition, we also explore other ways of distributing the $W = 6$ angles, such as continuously ("S-6") or randomly ("R-6"), respectively. We find our employed uniform placement is the best, since each training sample could cover the whole point cloud. In the following experiments, we use $W = 6$.

Finally, we explore the effect of the Z-dimensional unit Gaussian distribution. In Table. 4, we compare the results obtained with different Z, including $\{32, 64, 128, 256\}$. The results get better with increasing Z until $Z = 128$ while degenerating when Z is too big. We believe Z depends on the number of training samples, and both 64 and 128 are good for Z under ModelNet10. Z is set to 128 below.

Ablation study. We further explore how each module in MAP-VAE contributes to the performance. As shown in Table. 5, we remove a loss each time to highlight the corresponding module. The degenerated results indicate that all elements contribute to the discriminability of learned features, and self-reconstruction ("No R") contributes more than the half-to-half prediction ("No P").

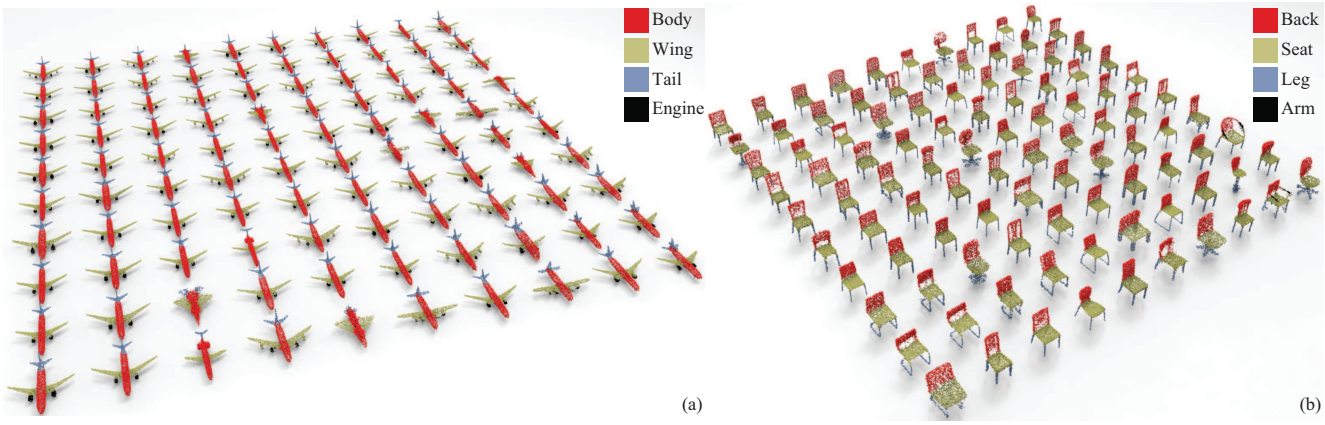

Figure 6. We show segmentation results from the airplane class in (a) and the chair class in (b).

Table 5. Ablation study,$\alpha = 0.01$,$\beta = 1000$,$W = 6$,$Z = 128$.

	No R	No P	No KL	All	AE	VAE	Eucli
%	91.63	92.40	93.17	**94.82**	92.29	93.28	93.61

In addition, we highlight our half-to-half prediction by showing the results obtained only by the pre-trained global encoder in Fig. 1 and this global encoder with a variational constraint (using the same balance weights as MAP-VAE), as shown by "AE" and "VAE". These results show that half-to-half prediction can help MAP-VAE understand point clouds better by leveraging effective local self-supervision. Moreover, the lower result of "Euclid" also indicates geodesic distance is superior to Euclidean distance to obtain semantic front half in the splitting of a point cloud.

Shape classification. We evaluate MAP-VAE in shape classification by comparing it with state-of-the-art methods under ModelNet40 [37] and ModelNet10 [37]. All the compared methods perform unsupervised 3D feature learning while using various 3D representations, including voxels, views and point clouds. As shown in Table 6, MAP-VAE obtains the best performance among these methods under ModelNet10. We employ the same parameters involved in Table 5 to produce our result under ModelNet40, where MAP-VAE also outperforms all point cloud based methods. Although view-based VIPGAN is a little better than ours, it cannot generate 3D shapes. These results indicate that MAP-VAE learns more discriminative global features for point clouds with the ability of leveraging more effective local self-supervision. Note that the results of LGAN, FNet and NSampler are trained under a version of ShapeNet55 that contains more than 57,000 3D shapes. However, there are only 51,679 3D shapes from ShapeNet55 that are available for public download. Therefore, MAP-VAE cannot be trained under the same number of shapes. To perform fair comparison, we use the codes of LGAN and FNet to produce their results under the same shapes in ModelNet, as shown by "LGAN(MN)" and "FNet(MN)".

Table 6. The classification accuracy (%) comparison among unsupervised 3D feature learning methods under ModelNet40 and ModelNet10. $\alpha = 0.01$,$\beta = 1000$,$W = 6$,$Z = 128$.

Methods	Modality	MN40%	MN10%
T-L Network[7]	Voxel	74.40	-
Vconv-DAE[30]	Voxel	75.50	80.50
3DGAN[36]	Voxel	83.30	91.00
VSL[24]	Voxel	84.50	91.00
VIPGAN[16]	View	91.98	94.05
LGAN[1]	Points	85.70	95.30
LGAN[1](MN)	Points	87.27	92.18
NSampler[28]	Points	88.70	95.30
FNet[40]	Points	88.40	94.40
FNet[40](MN)	Points	84.36	91.85
MRTNet[6]	Points	86.40	-
3DCapsule[42]	Points	88.90	-
PointGrow[33]	Points	85.80	-
PCGAN[21]	Points	87.80	-
Our	Points	**90.15**	**94.82**

Shape segmentation. We evaluate the local features learned by MAP-VAE for each point in shape segmentation. The ShapeNet part dataset [26] is employed in this experiment, where point clouds in 16 shape classes are involved to train MAP-VAE with the same parameters in Table 6.

We first extracted the learned feature of each point from the second-last layer in the global decoder D. Extracting the feature of each single point in the decoding procedure represents the ability of MAP-VAE to understand shapes locally at each point. Second, we map the ground truth label of each point to the reconstructed point cloud by voting 5 nearest labels for each reconstructed point. Third, we train a per-point softmax classifier under the training set, and test the classifier under the test set.

We use the same approach to obtain the results of the autoencoder in LGAN [1], and this autoencoder with a variational constraint. As the comparison shown in Table 7, our

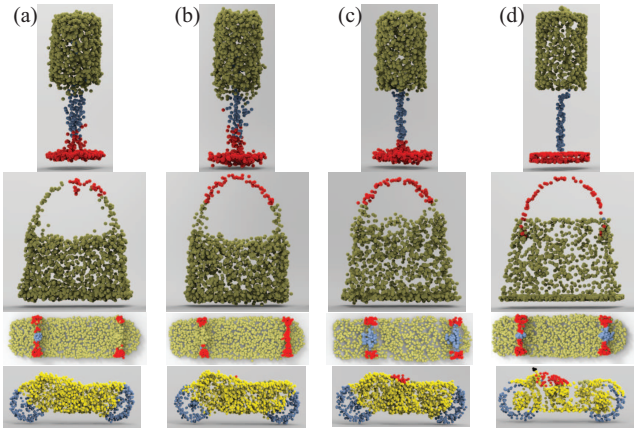

Figure 7. Segmentation comparison between "LGAN" (a), "L-GAN1" (b) in Table 7 and MAP-VAE (c). The ground truth is shown in (d). A color in the same row represents a part class.

results significantly outperform "LGAN" and "LGAN1" in terms of both mIoU and classification accuracy. We further visualize the segmentation comparison on four cases in Fig. 7. We find that the captured local geometry information helps MAP-VAE not only to reconstruct point clouds better, but also to learn more discriminative features for each point for better segmentation. Finally, we show 100 segmentation results in two challenging shape classes, respectively, i.e., airplane and chair, in Fig. 6. The consistent segmentation results also justify the good performance of MAP-VAE.

Shape generation. Next we demonstrate how to generate novel shapes using the trained MAP-VAE. Here, we first sample a noise vector from the employed Z-dimensional unit Gaussian distribution, and then, convey the noise to the global decoder D in branch R in Fig. 1.

Using MAP-VAE trained under ModelNet10 in Table 6, we generate some novel shapes in each of 10 shape classes in Fig. 10 (a), where we sample 4 noise vectors around the feature center of each shape class to generate 4 shape class specific shapes. The generated point clouds are sharp and with high fidelity, where more local geometry details are learned. Moreover, we also observe high quality point clouds in the same shape class from MAP-VAE trained under ShapeNet part dataset in Table 7. We generate 100 airplanes using 100 noise vectors sampled around the feature center of the airplane class, as demonstrated in Fig. 10 (b).

We further show the point clouds generated by the interpolated features between two feature centers of two shape classes, where the MAP-VAE trained under ModelNet10 in Table 6 is used. The interpolated point clouds are plausible and meaningful, and smoothly changed from one center to another center in Fig. 11 (a) and (b). Similar results can be observed by interpolations between two shapes in the same class in Fig. 11 (c) and (d), where the MAP-VAE trained under ShapeNet part dataset in Table 7 is used.

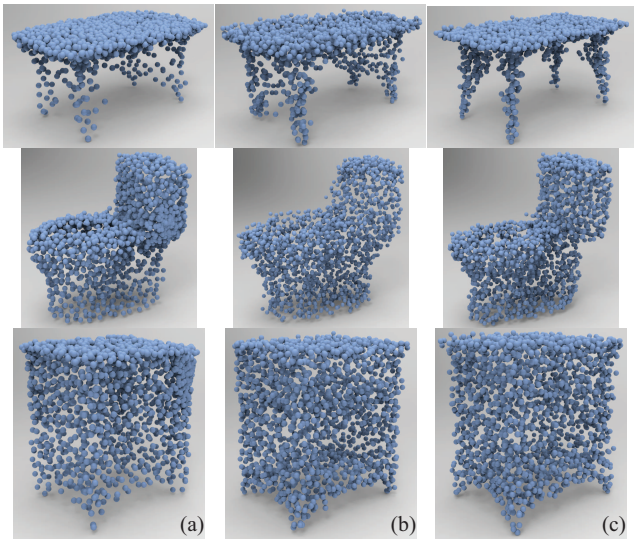

Figure 8. Compared to the three generated class centers under ModelNet10, MAP-VAE (in (c)) learns more local geometry details than "LGAN" (in (a)) and "LGAN1" (in (b)) in Table 7, due to the effective local self-supervision in half-to-half prediction.

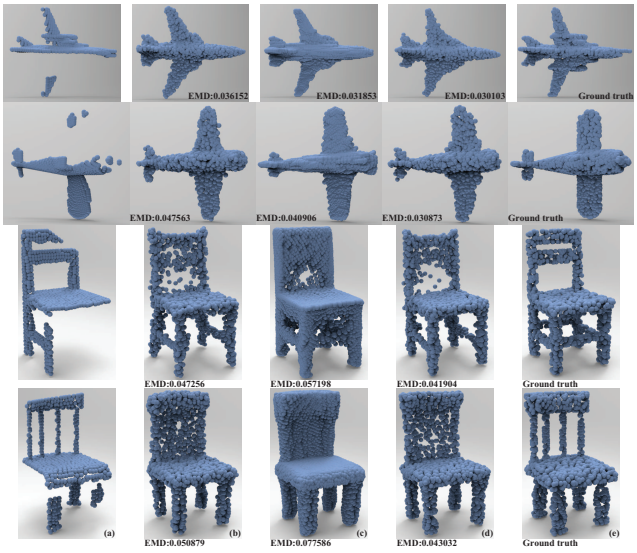

Figure 9. Visual comparison with "LGAN" (in (b)) in Table 7 and "PCN-EMD" (in (c)) in Table 8 for the completion of partial point clouds (a). MAP-VAE (in (d)) completes more geometry details.

Finally, we visually highlight the advantage of MAP-VAE by the point cloud generated at the feature center of a shape class. As demonstrated in Fig. 8, we compare MAP-VAE with the autoencoder in LGAN[1], and this autoencoder with a variational constraint. Using the trained decoder of each method, we generate a point cloud from the feature center at each of the three shape classes. Compared to the three class centers in Fig. 8 (a) and Fig. 8 (b), MAP-VAE in Fig. 8 (c) can generate point clouds with more local geometry details, such as sharp edges of parts.

Table 7. The segmentation comparison among unsupervised 3D feature learning methods under ShapeNet part dataset. The metric is mIoU(%) and per-point classification accuracy(%) on points. $\alpha = 0.01$,$\beta = 1000$,$W = 6$,$Z = 128$.

	Methods	Mean	Aero	Bag	Cap	Car	Chair	Ear	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate	Table
mIoU	LGAN [1]	57.04	54.13	48.67	62.55	43.24	68.37	58.34	74.27	68.38	53.35	82.62	18.60	75.08	54.70	37.17	46.71	66.39
	LGAN1 [1]	56.28	52.16	57.85	62.66	42.01	67.66	52.25	75.37	68.63	49.07	81.52	19.20	75.43	54.34	35.09	41.48	65.73
	Ours	**67.95**	**62.73**	**67.08**	**72.95**	**58.45**	**77.09**	**67.34**	**84.83**	**77.07**	**60.89**	**90.84**	**35.82**	**87.73**	**64.24**	**44.97**	**60.36**	**74.75**
ACC	LGAN [1]	78.24	74.93	84.36	77.02	71.10	78.23	78.34	84.41	78.29	69.05	86.86	67.93	90.42	81.95	68.44	82.27	78.25
	LGAN1 [1]	77.35	73.64	84.05	75.93	69.82	77.35	77.45	83.72	78.10	68.45	85.85	66.06	89.69	81.43	67.59	81.10	77.33
	Ours	**87.45**	**83.50**	**93.79**	**86.12**	**83.28**	**87.03**	**88.08**	**93.15**	**86.66**	**79.31**	**94.89**	**77.37**	**98.86**	**90.51**	**77.14**	**93.21**	**86.25**

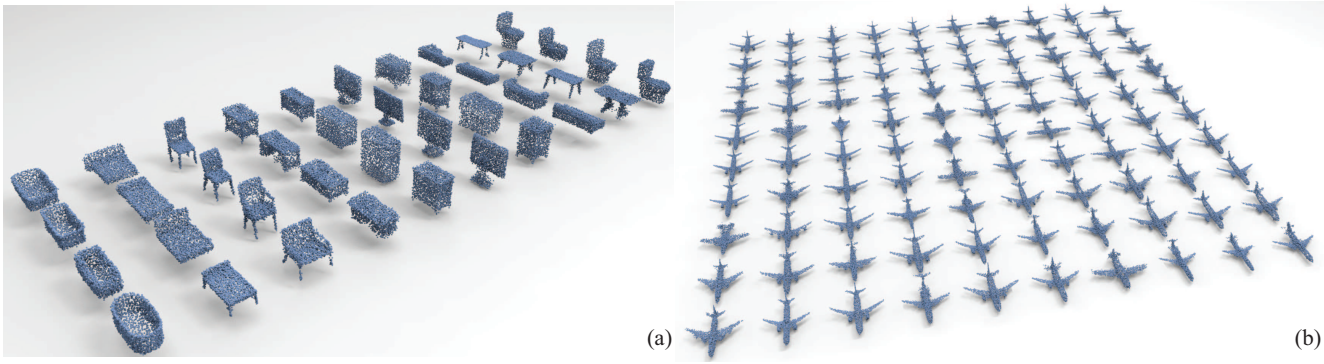

Figure 10. High fidelity novel shape generation by MAP-VAE trained under ModelNet10 in (a) and ShapeNet part dataset in (b).

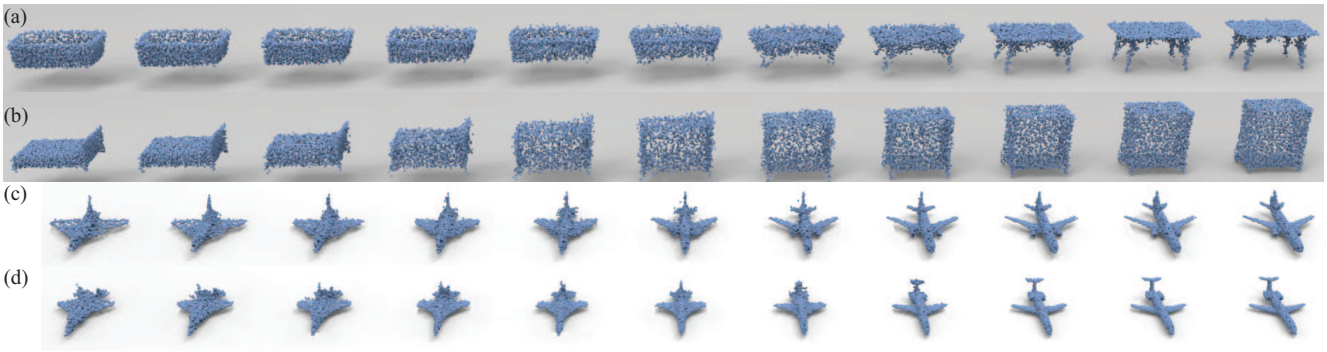

Figure 11. We show shape interpolation results between two different shape classes under ModelNet10 in (a) and (b), and the shape interpolation results between two shapes in the same class under the ShapeNet part dataset in (c) and (d).

Table 8. The completion comparison under airplane and chair classes in terms of EMD/point, $\alpha = 0$,$\beta = 1000$,$W = 12$,$Z = 0$.

Class	EPN[3]	Folding[40]	PCN-CD[41]	PCN-EMD[41]	LGAN[1]	Our
Airplane	0.061960	0.156438	0.046637	0.038752	0.033218	**0.032328**
Chair	0.076802	0.297427	0.086787	0.068074	0.055908	**0.055696**

Point cloud completion. MAP-VAE can also be used in point cloud completion, where we set $W = 12$ and remove the KL loss for fair comparison. We evaluate our performance under the training and test sets of partial point clouds in two challenging shape classes in [3], i.e., airplane and chair, where we employ the complete point clouds in [26] as ground truth. Since each partial point cloud has different number of points, we resample 2048 points to obtain the front and back halves. We compare with the state-of-the-art methods in Table 8. The lowest EMD distance shows that MAP-VAE outperforms all competitors. In addition, we also visually compare the completed point clouds in Fig. 9, where MAP-VAE completes more local geometry details.

7. Conclusions

We propose MAP-VAE for unsupervised 3D point cloud feature learning by jointly leveraging local and global self-supervision. MAP-VAE effectively learns local geometry and structure on point clouds from semantic local self-supervision provided by our novel multi-angle analysis. The outperforming results in various applications show that MAP-VAE successfully learns more discriminative global or local features for point clouds than state-of-the-art.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. J. Guibas. Learning representations and generative models for 3D point clouds. In *The International Conference on Machine Learning*, pages 40–49, 2018.

[2] K. Crane, C. Weischedel, and M. Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11):90–99, 2017.

[3] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] H. Deng, T. Birdal, and S. Ilic. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *Proceedings of European Conference on Computer Vision*, volume 11209, pages 620–638, 2018.

[5] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2463–2471, 2017.

[6] M. Gadelha, R. Wang, and S. Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, 2018.

[7] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of European Conference on Computer Vision*, pages 484–499, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

[9] Z. Han, X. Liu, Y.-S. Liu, and M. Zwicker. Parts4Feature: Learning 3D global features from generally semantic parts in multiple views. In *IJCAI*, 2019.

[10] Z. Han, Z. Liu, J. Han, C. Vong, S. Bu, and C. Chen. Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy. *IEEE Transactions on Cybernetics*, 49(2):481–494, 2019.

[11] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. Chen. 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):2268 – 2281, 2017.

[12] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li. Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine. *IEEE Transactions on Image Processing*, 25(11):5331–5344, 2016.

[13] Z. Han, Z. Liu, C. Vong, Y.-S. Liu, S. Bu, J. Han, and C. Chen. 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):3049–3063, 2018.

[14] Z. Han, Z. Liu, C.-M. Vong, Y.-S. Liu, S. Bu, J. Han, and C. Chen. BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation. *IEEE Transactions on Image Processing*, 26(8):3707–3720, 2017.

[15] Z. Han, H. Lu, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. P. Chen. 3D2SeqViews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999, 2019.

[16] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker. View Inter-Prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *AAAI*, 2019.

[17] Z. Han, M. Shang, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. P. Chen. SeqViews2SeqLabels: Learning 3D global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):1941–0042, 2019.

[18] Z. Han, M. Shang, X. Wang, Y.-S. Liu, and M. Zwicker. Y2Seq2Seq: Cross-modal representation learning for 3D shape and text by joint reconstruction and prediction of view and word sequences. In *AAAI*, 2019.

[19] Z. Han, X. Wang, C.-M. Vong, Y.-S. Liu, M. Zwicker, and C. P. Chen. 3DViewGraph: Learning global features for 3d shapes from a graph of unordered views with attention. In *IJCAI*, 2019.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[21] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov. Point cloud GAN. *CoRR*, abs/1810.05795, 2018.

[22] J. Li, B. M. Chen, and G. H. Lee. SO-Net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9397–9406. IEEE Computer Society, 2018.

[23] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on x-transformed points. In *NeurIPS*, pages 828–838, 2018.

[24] S. Liu, C. L. Giles, and A. G. O. II. Learning a hierarchical latent-variable model of 3D shapes. In *2018 International Conference on 3D Vision (3DV)*, 2018.

[25] X. Liu, Z. Han, Y. Liu, and M. Zwicker. Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. *AAAI*, 2019.

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

[28] E. Remelli, P. Baque, and P. Fua. NeuralSampler: Euclidean point cloud auto-encoder and sampler. *CoRR*, abs/1901.09394, 2019.

[29] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[30] A. Sharma, O. Grau, and M. Fritz. VConv-DAE: Deep volumetric shape learning without object labels. In *Proceedings of European Conference on Computer Vision*, pages 236–250, 2016.

[31] Y. Shen, C. Feng, Y. Yang, and D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4548–4557, 2018.

[32] M. Shoef, S. Fogel, and D. Cohen-Or. PointWise:an unsupervised point-wise feature learning network. *CoRR*, abs/1901.04544, 2019.

[33] Y. Sun, Y. Wang, Z. Liu, J. E. Siegel, and S. E. Sarma. PointGrow: Autoregressively learned point cloud generation with self-attention. *CoRR*, abs/1810.05591, 2018.

[34] D. Valsesia, G. Fracastoro, and E. Magli. Learning localized generative models for 3D point clouds via graph convolution. In *International Conference on Learning Representations*, 2019.

[35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829, 2018.

[36] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90. 2016.

[37] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[38] S. Xie, S. Liu, Z. Chen, and Z. Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018.

[39] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In *ECCV*, volume 11212, pages 90–105, 2018.

[40] Y. Yang, C. Feng, Y. Shen, and D. Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[41] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert. PCN: Point completion network. In *Proceedings of 2018 International Conference on 3D Vision*, 2018.

[42] Y. Zhao, T. Birdal, H. Deng, and F. Tombari. 3D pointcapsule networks. *CoRR*, abs/1812.10775, 2018.