ITC 4/51

Information Technology and Control
Vol. 51 / No. 4 / 2022
pp. 723-737
DOI 10.5755/j01.itc.51.4.29894

UPSNet: Universal Point Cloud Sampling Network Without Knowing Downstream Tasks

Received 2021/09/27 | Accepted after revision 2022/02/21

http://dx.doi.org/10.5755/j01.itc.51.4.29894

HOW TO CITE: Tian, F., Song, Y., Jiang, Z., Tao, W., Jiang, G. (2022). UPSNet: Universal Point Cloud Sampling Network Without Knowing Downstream Tasks. *Information Technology and Control*, 51(4), 723-737. http://dx.doi.org/10.5755/j01.itc.51.4.29894

UPSNet: Universal Point Cloud Sampling Network Without Knowing Downstream Tasks

Fujing Tian, Yang Song, Zhidi Jiang, Wenxu Tao, Gangyi Jiang
Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China

Corresponding author: jianggangyi@nbu.edu.cn

With the development of three-dimensional sensing technology, the data volume of point cloud grows rapidly. Therefore, point cloud is usually down-sampled in advance so as to save memory space and reduce the computational complexity for its downstream processing tasks such as classification, segmentation, reconstruction in learning based point cloud processing. Obviously, the sampled point clouds should be well representative and maintain the geometric structure of the original point clouds so that the downstream tasks can achieve satisfied performance based on the point clouds sampled from the original ones. Traditional point cloud sampling methods such as farthest point sampling and random sampling mainly heuristically select a subset of the original point cloud. However, they do not make full use of high-level semantic representation of point clouds, are sensitive to outliers. Some of other sampling methods are task oriented. In this paper, a Universal Point cloud Sampling Network without knowing downstream tasks (denoted as UPSNet) is proposed. It consists of three modules. The importance learning module is responsible for learning the mutual information between the points of input point cloud and calculating a group of variational importance probabilities to represent the importance of each point in the input point cloud, based on which a mask is designed to discard the points with lower importance so that the number of remaining points is controlled. Then, the regional learning module learns from the input point cloud to get the high dimensional space embedding of each region, and the global feature of each region are obtained by weighting the high dimensional space embedding with the variational importance probability. Finally, through the coordinate regression module, the global feature and the high dimensional space embedding of each region are cascaded for learning to obtain the sampled point cloud. A series of experiments are implemented in which the point cloud classification, segmentation, reconstruction and retrieval are performed on the reconstructed point clouds sampled with different point cloud sampling methods. The experimental results show that the proposed UPSNet can provide more reasonable sampling result of the input point cloud for the downstream tasks of classification, segmentation, reconstruction and retrieval, and is superior to the existing sampling methods without knowing the downstream tasks. The proposed UPSNet is not oriented to specific downstream tasks, so it has wide applicability.

KEYWORDS: Point cloud, universal point cloud sampling network, variational importance probability, downstream tasks.

1. Introduction

With the development of three-dimensional (3D) sensing technology, more and more attentions are paid to point cloud processing and its applications. Point cloud is a set of unordered points in 3D space, and each point has 3D coordinate, additionally with other attributes including color, normal, etc. Point cloud has wide applications in various fields, such as autonomous driving [6, 11, 12], surveying and mapping, 3D reconstruction [14, 27, 36, 37], virtual reality and augmented reality, etc. The original point cloud is with huge amount of data, which not only requires a lot of memory space, but also leads to the increase of computational complexity of subsequent point cloud processing. Therefore, point cloud down sampling is usually performed in advance before the tasks of classification, segmentation, reconstruction, retrieval, compression, and so on [13, 28, 34].

Point cloud down sampling can be usually described as finding a subset $\boldsymbol{Q}$ of a given 3D unordered point set $\boldsymbol{P}$, where $\boldsymbol{P}=\{\boldsymbol{P}_i|i=1,...,n\}$, $\boldsymbol{Q}=\{\boldsymbol{Q}_i|i=1,...,k\}$, $\boldsymbol{Q}\subset\boldsymbol{P}$. Obviously, the point cloud sampling method should retain the points that can well reflect the geometric structure of the original point cloud as much as possible under the constraint of the number of sampling points.

The classical point cloud sampling methods mainly include clustering based sampling [2, 33], iterative based sampling [20, 30, 32] and formula based sampling [3, 10, 17]. Benhabiles et al. [2] used volume clustering method to generate coarse point cloud, which accelerated the sampling speed. Yu et al. [33] used local clustering and hierarchical clustering to sample point clouds. Yang et al. [30] redefined the mean curvature of points using principal component analysis and Fourier transform, and then iteratively deleted the points with low mean curvature. Song et al. [20] defined the importance of a point by using the normal vector information and the distance between the point and its neighborhood, then deleted the point with the least importance, and updated the importance and normal vector of the other points iteratively. Leal et al. [10] first clustered the points to retain the points with high curvature, and then selected a reduced set whose density is equivalent to the original point set from the remaining points. Qi et al. [17] formulated point cloud simplification as a trade-off between preserving sharp features and keeping uniform density during resampling, and retained the sharp and uniform features of the point cloud through the graph filter. Dinesh et al. [3] derived a sampling objective that maximizes the stability (maximizes the smallest eigenvalue $\boldsymbol{\lambda}_{\mathbf{min}}(\mathbf{B})$ of a coefficient matrix $\mathbf{B}=\mathbf{H}^{\mathrm{T}}\mathbf{H}+\boldsymbol{\mu}\mathbf{L}$) of a linear system super-resolving a sub-sampled point cloud. The classical point cloud down sampling methods usually focus on maintaining geometric structure of the input point clouds, but ignore the high-level features of point cloud, hence their generalization ability is limited, and the sampled subsets of the input point clouds may be not most suitable for the downstream tasks.

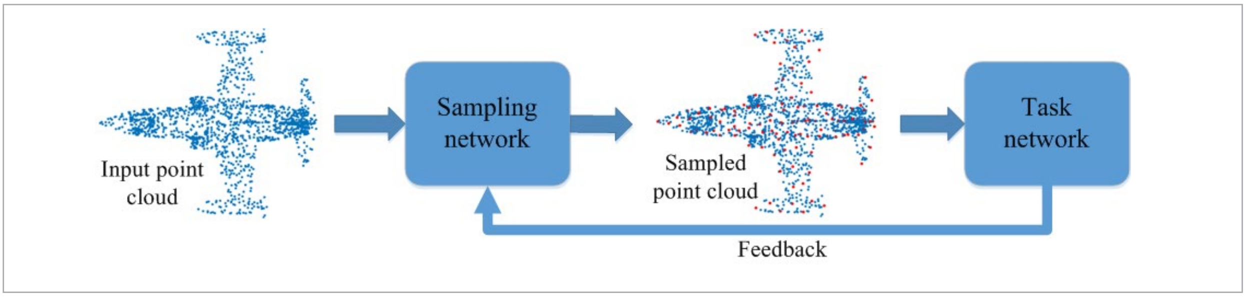
In recent years, many learning based methods are proposed for point cloud processing. Since the sampled points are desired to keep the geometric structure of the original point cloud, the procedure of point could sampling can be regarded as a kind of feature learning in point cloud processing so as to exploit the most significant points for feature extraction and transfer them to the downstream tasks. Most efforts have been made on designing deep learning network to learn the features of point cloud directly. Qi et al. [15] designed a well-known network, that is, PointNet, to directly learn unordered point clouds. It has achieved good results in the task of point cloud classification and segmentation by learning the features of each point individually and using a symmetric max pooling function to keep the point's permutation invariance. However, PointNet does not consider the correlation between points and ignores the local features of point clouds. In view of this defect, they further designed the PointNet++ [16], in which the point cloud subset is firstly obtained through farthest point sampling (FPS), and then taking the subset as the center point, neighborhood points within a certain range are queried in the original point cloud, so that a multi-scale spherical neighborhood can be constructed to capture multi-level local information of the point cloud. Wang et al. [25] designed a dynamic graph CNN (DGCNN) by computing the k-nearest neighbors of each point to aggregate the features of different levels in each local region. Taghanaki et al. [23] proposed the PointMask with regularization term to mask irrelevant input variables. In recent years, many point cloud learning networks have been constructed for other tasks, such as point cloud segmentation [21, retrieval [24], registration [1], reconstruction [22], object recognition [9], and so on. Although these networks are designed for different tasks, the used sampling strategies are static, which are only based on the spatial location of the input point cloud, but do not use the high-level representations of the point cloud.

Farthest point sampling and random sampling (RS) are two popular sampling methods used in learning based networks for point cloud processing. They are heuristic and able to keep the geometric structure of the object by iteratively collecting samples on the surface of the object. FPS uses Euclidean distance metric to iteratively search for the sampling points, and the selected point is that farthest from other unselected members in each iteration [16]. Kulikajevas et al. [8] proposed a two-tiered deep neural network for self-occluding humanoid pose reconstruction, in which the clipping network is designed to clip the region of interest and down sampling it with FPS for the subsequent reconstruction network. Since FPS can well cover the whole set of points, several methods use it to extract the feature point [35, 38]. However, due to the high computational complexity, FPS is generally suitable for small-scale point clouds. RS does sampling from the probability distributions. Hu et al. [7] used RS for semantic segmentation of large-scale point clouds. Although RS has excellent computing and memory efficiency, it may discard key features. Therefore, they designed a local feature aggregation module to increase the receptive field for each 3D point so as to preserve the geometric details. These typical sampling methods are mainly designed based on low dimensional Euclidean space, and do not make full use of point cloud's high-level semantic representation. Moreover, they are quite sensitive to outliers [29], which may degrade the performance of the downstream tasks due to inappropriate sampling of the input original point clouds.

Some studies have focused on the design of point cloud sampling network based on downstream tasks [4, 18, 39], which are also called as task oriented sampling in [18]. Zhu et al. [39] presented a point rank sampling method in their shape completion network to rate and sample feature points more objectively through local outline form. Dovrat et al. [4] proposed a data driven point cloud sampling network namely S-Net. They firstly used PointNet to obtain the global feature of the input point cloud, and then used full connection layer regression to generate the sampled subset of the input point cloud. By freezing the downstream task network, S-Net can generate a sampled subset of the input point cloud that is more conducive to network learning of the downstream task such as point cloud classification, retrieval and reconstruction. Considering that S-Net directly reconstructs point clouds from global features without using the correlation between points, Qian et al. [18] explored the local geometric correlation of point cloud from the perspective of matrix optimization, and proposed the MOPSNet to transform 3D points into high dimensional feature space and construct a constrained differentiable matrix optimization problem with implicit objective function. By simulating the problem of matrix optimization, the sampled subset of input point cloud is obtained. S-Net and MOPSNet are the task oriented sampling network which is simply described in Figure 1 where the red points construct the sampled subset $\boldsymbol{Q}$ of the input point cloud $\boldsymbol{P}$. In such kind of methods, the relevant downstream tasks are first scheduled, and downstream task network is trained in advance, then the trained downstream task network is used to feedback and adjust the sampling network, so that the generated points (that is, the sampled subset $\boldsymbol{Q}$ of the input point cloud) can better adapt to the downstream tasks. However, sometimes downstream tasks may be unknown in advance. In such cases, the sampling network is desired to adapt to most downstream tasks, moreover, it is better to be unsupervised and do not need feedback regulation of downstream tasks. Therefore, it is necessary to design universal

Figure 1

Downstream task oriented sampling network



point cloud sampling network independent of downstream tasks.

Different from the previous task oriented sampling networks, this paper proposes an universal point cloud sampling network (called as UPSNet), which can sample representative points without knowing the downstream tasks in advance and feedback adjustment. The main contributions of this paper are as follows.

1 An end-to-end universal point cloud sampling network namely UPSNet is proposed, which can better adapt to the cases of unknown downstream tasks and have better versatility.

2 An importance learning module is designed to improve the performance of sampling, it can learn the variational importance probability of each point in input point cloud through the re-parameterization technique. By shielding the points with low importance probability, the proposed network can learn the feature of the points with higher importance which is more beneficial to the downstream tasks.

3 To enhance the learning ability of the local features of the point cloud in the proposed UPSNet, a regional learning mechanism is designed to realize the sub-region learning of the point cloud, which can learn the high dimension space embedding of each region and help the network reconstruct the point cloud.

The rest of this paper is organized as follows. Section II describes the proposed UPSNet in detail. Section III analyzes the experimental results of the proposed UPSNet and its performance in point cloud classification, segmentation, reconstruction and retrieval. Finally, Section IV concludes the paper.

2. Proposed UPSNet

In this paper, a universal point cloud sampling network (UPSNet) which can be trained independently of downstream task is proposed. Its framework is given in Figure 2. The UPSNet is mainly composed of three modules, including importance learning module, regional learning module and coordinate regression module, as shown in Figure 2(a). Firstly, the importance of each point in high dimensional space is obtained by the importance learning module. Then, the high dimensional features of each region are learned in the regional learning module. Finally, the sampled subset of the input point cloud is reconstructed in coordinate regression module.

2.1. Importance Learning Module

The purpose of point cloud sampling is to construct its representative subset. Because all points in the point cloud may have different importance, if all points are equally learned, the selected point cloud subset may not be representative. Therefore, we analyze the importance of point to point from the perspective of mutual information, and design the importance learning module. Let $\boldsymbol{P}=\{\boldsymbol{P}_i|i=1,...,n\}$ denote a 3D unordered point set, $\boldsymbol{P}_i\in\mathbb{R}^d$. Generally, $d=3$ is taken as the dimension of coordinate of 3D space. The goal is to calculate a set of variational importance probabilities $J=\{J_i|i=1,...,n\}$ corresponding to $\boldsymbol{P}$ by optimizing

$$L(\omega) = I(J,\boldsymbol{Q};\omega) - \beta I(J,\boldsymbol{P};\omega)\,, \quad (1)$$

where $I(\cdot)$ denotes mutual information estimator, ω is network parameters, β is a scalar weight, and $\boldsymbol{Q}$ is the sampled point cloud of $\boldsymbol{P}$, $\boldsymbol{Q}\subset\boldsymbol{P}$.

Assuming that J obeys normal distribution, $J \sim N(\mu, \sigma^2)$, the mean μ and standard deviation σ of distribution parameters are obtained by the network learning. We use the re-parameterization technique to get J, $J=\mu+\sigma\varepsilon$, $\varepsilon \sim N(0,1)$. Then, a mask M is designed for point screening according to the number of points to be sampled, denoted by

$$M = \mathrm{ReLU}(\sigma(J) - T_h)\,. \tag{2}$$

Let α be the importance of each point in the point cloud, and the higher the value is, the more important the feature of the point is. The mask operation can be used to discard the points whose importance is lower than a threshold T_h, so that the number of remaining points is equal to that of the sampling points. The specific process of the importance learning module is shown in Figure 2(b). Note that the importance α will also be used as the weight to get the global feature of each region in regional learning module.

As an example, Figure 3 shows the sampling results of the mask operation under different thresholds. In Figure 3, the red points indicate the remaining

Figure 2(a)

Framework of the proposed universal point cloud sampling network (UPSNet)

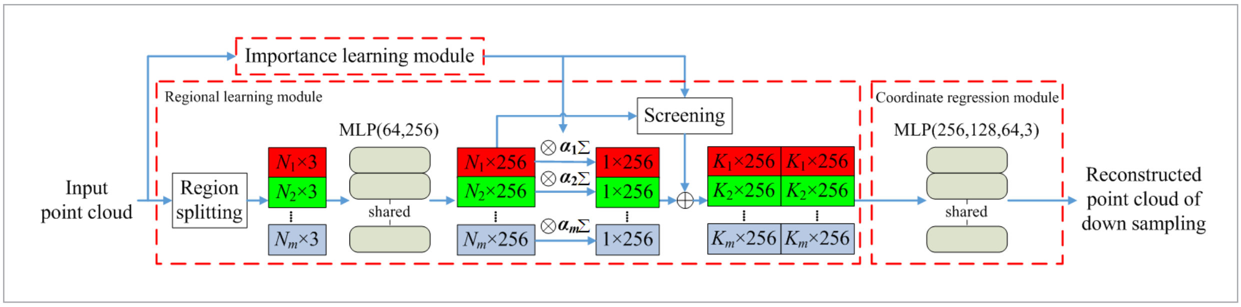


Figure 2(b)

Framework of the importance learning module in the proposed UPSNet

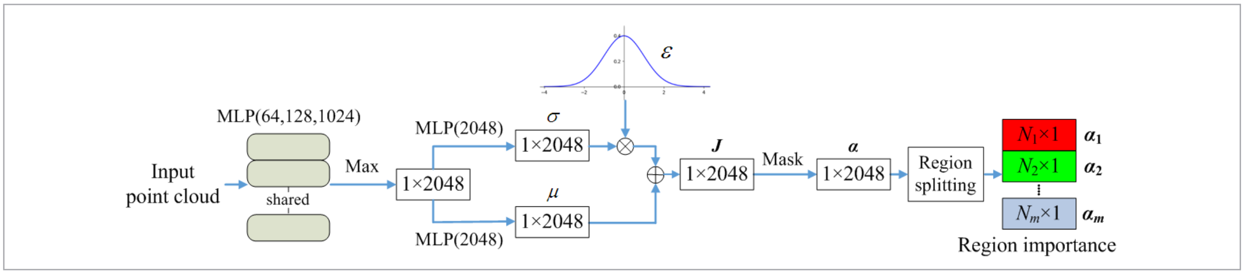
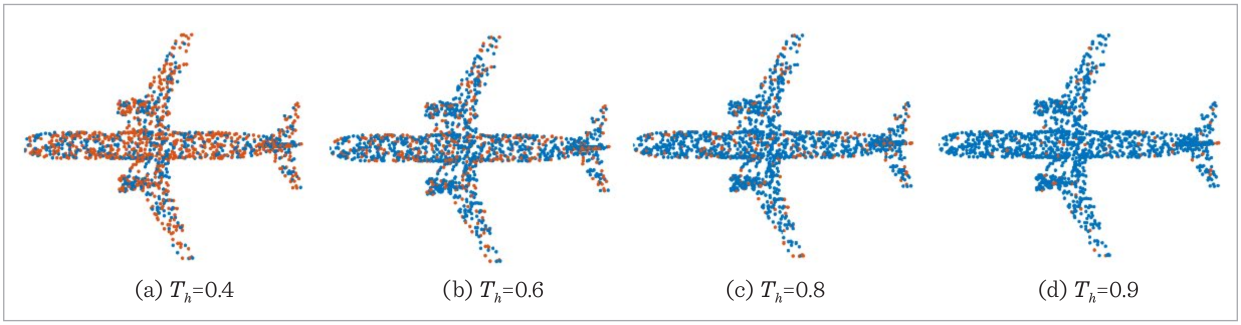


Figure 3

Remaining points after mask operation with different thresholds


(a) T_h=0.4 (b) T_h=0.6 (c) T_h=0.8 (d) T_h=0.9

points, while the discarded ones are marked in blue. In the process of network learning, the network is more desired to learn features from the points that can represent the contour of the object. From Figure 3, it is seen that as the threshold T_h increases, the number of remaining points decreases, and the mask operation tends to select the points that can represent the shape of the object, as shown in Figure 3(d). This is beneficial to describing the global feature of the object. In the experiment, the threshold T_h can be used to determine the number of the sampled points from the input point cloud, so that the remaining points are just equal to the number of points we need to sample.

2.2. Regional Learning Module

It is believed that the learning network should not only reconstruct the entire point cloud, but also reconstruct the fine-grained components of the point cloud. Therefore, it is needed to design the corresponding regional learning module for all local regions.

For the input point set $\boldsymbol{P}=\{\boldsymbol{P}_i|i=1,...,n\}$, suppose it can be segmented to m regions, denoted by $\boldsymbol{R}=\{\boldsymbol{R}_i|i=1,...,m\}$, where the region $\boldsymbol{R}_i$ has totally N_i points, satisfying $N_1+N_2+...+N_m=n$. By learning the feature mapping function f: $\mathbb{R}^3\rightarrow\mathbb{R}^d$ and mapping each region to a high dimensional space, the corresponding high dimensional space embedding $\boldsymbol{F}=\{\boldsymbol{F}_i|i=1,...,m\}$ of each region is obtained by multi-layer perceptron (MLP). The feature mapping function f is defined by

$$f(\boldsymbol{P}) = MLP(\boldsymbol{R}_1 \,||\, \boldsymbol{R}_2 \,||\, ... \,||\, \boldsymbol{R}_m), \quad (3)$$

where $MLP(\cdot)$ is an MLP, and || denotes cascade indicating the connection of different regions for learning. In this way, each region is connected to facilitate the subsequent extraction of weighted features of different regions.

Let $\boldsymbol{\alpha}_i$ denote the regional weight of the i-th region, $\boldsymbol{\alpha}_i=\{\alpha_i^j\,|j=1,...,N_i\}$, and α_i^j be the weight of the j-th point in the i-th region which is in fact the importance of the point obtained in the importance learning module. Then the regional weight $\boldsymbol{\alpha}=\{\alpha_i|i=1,...,m\}$ is multiplied with the corresponding high dimensional space embedding $\boldsymbol{F}$ of each region and the products are summed, so as to obtain the weighted feature G_i of the i-th region, denoted by

$$G_i = \sum_{j=1}^{N_i} \alpha_i^j \cdot F_i^j, \quad (4)$$

where F_i^j is feature of the j-th point in the i-th region and $\boldsymbol{F}_i=\{F_i^j\,|j=1,...,N_i\}$, G_i of all regions constitute $\boldsymbol{G}=\{G_i|i=1,...,m\}$. Here, considering that the maximum pooling cannot fully represent the features of a region which is composed of many points, the weighting method in Equation (4) is used to obtain the global features instead of the maximum pooling. The weighting method takes into account the features of all points and weights the features according to the importance of each point, hence more representative features of the local region can be obtained.

Then, for each region $\boldsymbol{R}_i$, the points whose importance is greater than the threshold T_h are selected, and the sampled subset of the input point cloud $\boldsymbol{S}=\{\boldsymbol{S}_i|i=1,...,m\}$ is obtained, $\boldsymbol{S}_i\subset\boldsymbol{R}_i$, where the region $\boldsymbol{S}_i$ has totally K_i points, satisfying $K_1+K_2+...+K_m=k$ and k is the total number of sampling points. The global feature G is cascaded with the high dimensional space embedding F of each region in the point cloud subset to obtain the final feature $\boldsymbol{T}=\{T_i|i=1,...,m\}$ as follows:

$$T_i = F_i \,||\, G_i\,. \quad (5)$$

2.3. Coordinate Regression Module

Through the above two modules, the designed network obtains the subset $\boldsymbol{S}$ of point cloud and the corresponding final feature $\boldsymbol{T}$ of each point. In coordinate regression module, MLP is used to regress the point cloud subset $\boldsymbol{S}$ in high dimensional space to the 3D spatial coordinate system to get the output of the proposed UPSNet. Considering that the generated point cloud is not necessarily a subset of the input point cloud, the nearest point matching strategy in S-Net [34] is adopted in the proposed point cloud sampling network to match the generated point cloud with the input point cloud to get the final point cloud subset.

2.4. Joint Loss Function in the Proposed UPSNet

In order to train UPSNet better, joint loss function is presented to train UPSNet from end to end. The joint loss function consists of reconstruction loss and

Kullback Leibler divergence. Based on the features of important points learned above, it is hoped that the network can reconstruct the point cloud subset close to the underlying surface of the point cloud according to these features so that the reconstructed subset can help the downstream tasks achieve good performance.

There are two kinds of reconstruction loss, earth mover's distance (EMD) [19] and Chamfer distance (CD) [5]. For two point clouds $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ with the same size, their EMD $L_{EMD}(\boldsymbol{P}_1, \boldsymbol{P}_2)$ is defined by

$$L_{EMD}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \min_{\phi:\boldsymbol{P}_1 \to \boldsymbol{P}_2} \sum_{x \in \boldsymbol{P}_1} \left\| x - \phi(x) \right\|_2 , \qquad (6)$$

where ϕ is double mapping.

EMD is differentiable everywhere, but it is computationally expensive and limits the size of the input set to be consistent. Therefore, the Chamfer distance is used as the reconstruction loss in this paper since it is more convenient to be calculated and has no requirement for input. The Chamfer distance is the square distance from each point in one set to its nearest neighbor in another set. For two point clouds $\boldsymbol{P}$ and $\boldsymbol{Q}$ with different points, their Chamfer distance L_{CD} is defined by

$$L_{CD}(\boldsymbol{P}, \boldsymbol{Q}) = \sum_{x \in \boldsymbol{P}} \min_{y \in \boldsymbol{Q}} \left\| x - y \right\|_2^2 + \sum_{y \in \boldsymbol{Q}} \min_{x \in \boldsymbol{P}} \left\| x - y \right\|_2^2 , \qquad (7)$$

where $\boldsymbol{P}$ is the input point cloud and $\boldsymbol{Q}$ is the reconstructed point cloud.

In this paper, Kullback Leibler divergence KL is used to measure the degree of similarity between the distribution of variational importance probability J and normal distribution. For variational importance probability $J=\mu+\varepsilon\sigma$, it is forced to obey normal distribution through sampling normal distribution $\varepsilon\sim N(0,1)$, where μ and σ are the mean and variance of the distribution. Thus, the divergence loss function L_{KL} between two distributions is defined by

$$L_{KL} = KL(p(J \mid P) \,\|\, r(J)) , \qquad (8)$$

where $r(J)$ is the variational approximation of marginal $p(J) = \int p(P)p(J \mid P)dP$.

Finally, by minimizing the following joint loss function L, the UPSNet is trained from end to end, and L is computed by

$$L = L_{CD} + L_{KL} . \qquad (9)$$

3. Experimental Results

To verify the effectiveness of the proposed UPSNet, a series of experiments including point cloud classification, segmentation, reconstruction and retrieval are performed on the reconstructed point clouds sampled with different comparative point cloud sampling methods. Random sampling (RS) and farthest point sampling (FPS) [16] are compared as the classical non-data driven sampling methods. Moreover, in order to compare with S-Net [4], we take its module that generates the point cloud as unsupervised S-Net (USNet). In addition, we also compare the sampling performance of the proposed UPSNet with that of the UPSNet(v) which does not consider regional learning, that is, the "Region splitting" in Figure 2 is omitted.

3.1. Network Training and Datasets

The proposed UPSNet is implemented on Pytorch, and GeForce RTX2080Ti GPU is used in the experiments. The input point cloud of the network has 2048 or 1024 points, while the point cloud reconstructed with the sampled subset has $2048/r$ or $1024/r$ points and r is the sampling rate. The proposed UPSNet uses RMSProp solver with the base learning rate of 0.001 and 400 epochs are trained with the batch size of 16.

ShapeNet [31]: The dataset contains 16881 models of 16 categories, and a total of 50 parts are annotated. Since the proposed UPSNet network needs regional learning, the models whose number of points of the annotated parts is less than 100 in the ShapeNet dataset are excluded in the experiments. In addition, considering the small number of some categories of models, we perform data augmentation on these categories of models through rotating them by 20 degree. Finally, there are totally 17897 models, and 80% of them are used as training set and the rest 20% as test set.

ModelNet40 [26]: This dataset contains a total of 12311 3D mesh models of 40 categories. It includes 9843 training models and 2468 test models. We uniformly sample 1024 points from the 3D mesh model and normalize them into the unit circle.

3.2. Test with the Downstream Task of Point Cloud Classification

In this subsection, two kinds experiments of point cloud classification are performed on ShapeNet dataset, and PointNet [15] is used as the benchmark network of classification. In *Classification* 1, the same original training set (not sampled) is used to train the classification network, and the classification accuracy is tested on the reconstructed test sets respectively obtained with different sampling methods so as to compare the effectiveness of different sampling methods. While in *Classification* 2, the classification network is trained with the reconstructed training sets respectively obtained with different sampling methods, and the classification accuracy is tested on the same original test set.

***Classification* 1:** The original training set of ShapeNet dataset (models with 2048 points) is used to train the classification network, that is, PointNet, and then the original test set (models with 2048 points) is used to generate the reconstructed test sets (models with 2048/r points) with respect to different sampling rate r through different sampling methods. Since the reconstructed point cloud is not necessarily a subset of the input data, we use the same matching strategy as in S-Net [4] to match the generated point cloud with the input point cloud so as to get the matched test set. then, the reconstructed test sets (models with 2048/r points) generated with different sampling methods are put into the trained PointNet respectively to get classification results and the corresponding classification accuracy is calculated to evaluate the effectiveness of the different sampling methods. The classification accuracy with respect to different sampling methods are shown in Table 1, where MA (Mean Accuracy of each category) represents the average accuracy of each category of point cloud and the best results are in bold.

In Table 1, the comparative point cloud sampling methods include random sampling (RS), farthest point sampling (FPS) [16], unsupervised S-Net (USNet) [4], the proposed UPSNet without regional learning module (denoted as UPSNet(v)) and the proposed UPSNet. It is seen that with the decrease of the number of sampling points (called sampling size), the classification accuracy of all sampling methods decreases. Compared with the sampling size of 1024, the classification accuracy of RS, FPS and USNet under the sampling size of 64 decreases by about 3%, while for UPSNet(v) and UPSNet they decrease by only 2.5% and 2%. Although UPSNet(v) considers the importance of each point in point cloud, it only utilizes the global features but no local features of each region, so the reconstructed point cloud sampled with UPSNet(v) may be farther from the underlying surface of the original point cloud compared with UPSNet. In contrast, the proposed UPSNet can generate more representative point cloud subsets by additionally learning the local features of each region so it achieves higher classification accuracy at low sampling size.

Table 1

Classification accuracy with respect to different sampling methods on ShapeNet dataset, where the original training set (models with 2048 points) is used to train the classification network but the reconstructed test sets (models with 2048/r points) generated with different sampling methods are respectively used for testing (MA: Mean Accuracy of each category) unit: %

Point cloud sampling method	Sampling size (2048/r)				
	1024	512	256	128	64
RS	96.2	96.6	96.5	95.3	93.2
FPS	97.1	96.9	96.5	95.4	94.1
USNet	**97.2**	**97.2**	96.9	96.1	93.5
UPSNet(v)	**97.2**	97.1	97.0	96.4	94.7
UPSNet	**97.2**	**97.2**	**97.1**	**96.5**	**95.2**

***Classification* 2:** In this comparative experiment, the classification network PointNet is respectively trained with different reconstructed training sets (models with 2048/r points) generated by different sampling methods, and then tested on the original test set (models with 2048 points) of ShapeNet dataset. The classification accuracy results with respect to different sampling methods are shown in Table 2. It is seen that when fewer points are sampled with RS, FPS and USNet, the classification network cannot learn good global features, so that when the original test set of point clouds with 2048 points are input into the classification network for testing, the global features obtained from the test set cannot well match the global features learned from the reconstructed training sets, leading to the lower classification accuracy. In contrast, the reconstructed training set sampled by the proposed UPSNet is more suitable for network learning, especially for large sampling rate r.

Table 2

Classification accuracy with respect to different sampling methods on ShapeNet dataset, where the classification network is respectively trained with the reconstructed training sets (models with 2048/*r* points) obtained by different sampling methods but the same original test set (models with 2048 points) is used for testing (MA: Mean Accuracy of each category) unit: %

Point cloud sampling method	Sampling size (2048/*r*)				
	1024	512	256	128	64
RS	97.0	96.9	96.0	93.1	87.1
FPS	**97.2**	**97.2**	**97.0**	96.4	92.7
USNet	97.1	97.1	96.8	96.0	93.4
UPSNet(v)	**97.2**	**97.2**	96.8	96.3	93.7
UPSNet	**97.2**	**97.2**	96.9	**96.7**	**94.3**

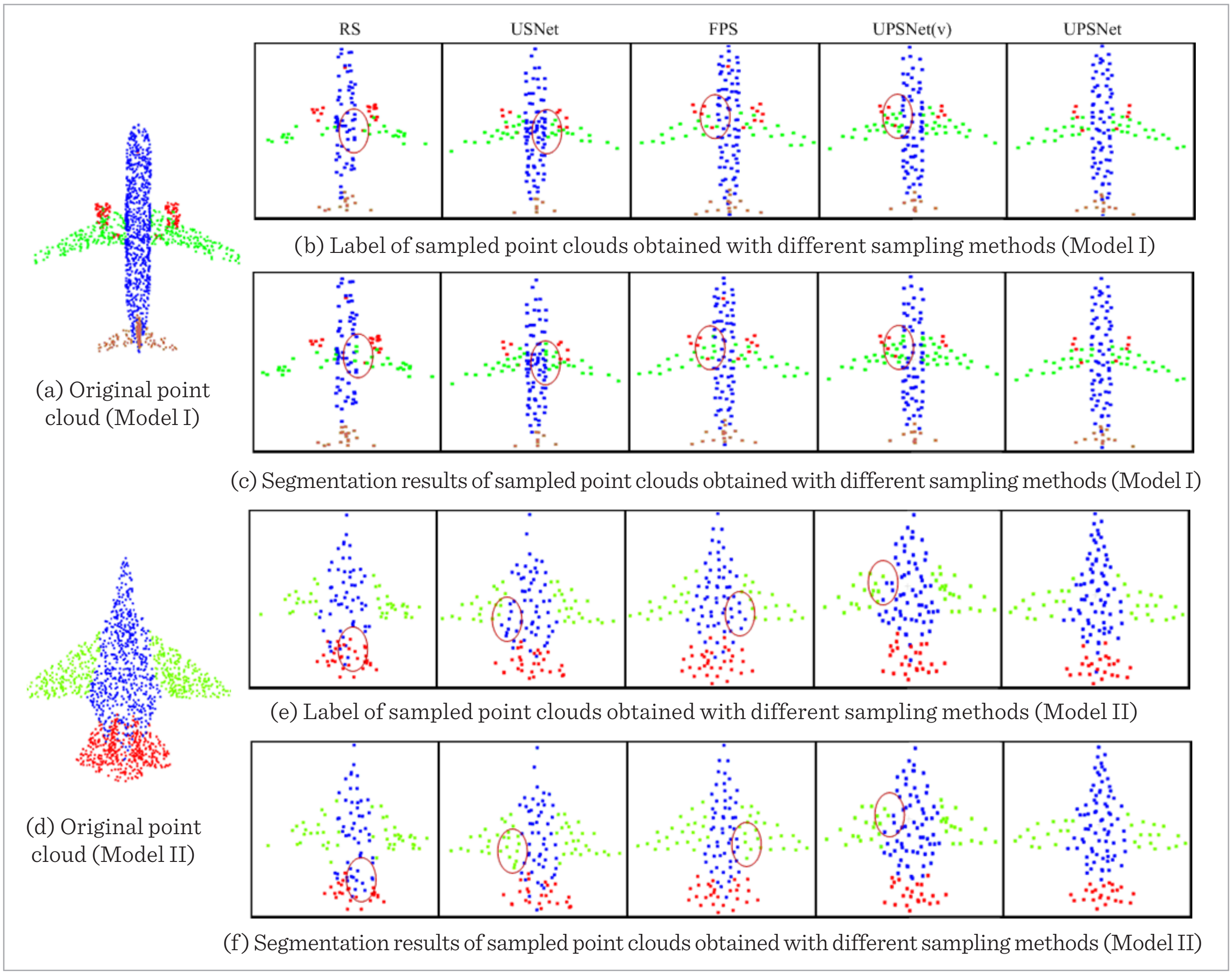
3.3. Test with the Downstream Task of Point Cloud Segmentation

3D point cloud segmentation is a challenging fine-grained recognition task, which needs to understand the role of each point in its own category. Therefore, it is necessary to have an effective sampling method, which can make the point cloud subset better express the characteristics of the input point cloud, so that the subsequent segmentation network can segment the point cloud more accurately.

Similar to the experiments of *Classification* 1, the experiments of segmentation task are also carried out on ShapeNet dataset. The training set (models with 2048 points) is used to train the sampling models such as the proposed UPSNet, and then the test set (models with 2048 points) is used to generate different recons-

Figure 4

Segmentation results of the reconstructed point clouds generated with different sampling methods at the sampling rate *r* 16



(a) Original point cloud (Model I)

(b) Label of sampled point clouds obtained with different sampling methods (Model I)

(c) Segmentation results of sampled point clouds obtained with different sampling methods (Model I)

(d) Original point cloud (Model II)

(e) Label of sampled point clouds obtained with different sampling methods (Model II)

(f) Segmentation results of sampled point clouds obtained with different sampling methods (Model II) tructed test sets (models with 2048/*r* points) through different sampling methods. The benchmark network used for point cloud segmentation is PointNet [15]. Firstly, the training set (models with 2048 points) of ShapeNet dataset is used to train the PointNet segmentation model, then the reconstructed test sets (models with 2048/*r* points) generated with different sampling methods are put into the trained PointNet respectively to calculate the segmentation accuracy with respect to different sampling methods, so that the effectiveness of these sampling methods under the task of point cloud segmentation can be evaluated.

The results of segmentation accuracy with respect to different sampling methods are shown in Table 3 where mIoU (mean Intersection over Union) of point clouds is used as the index and the best results are in bold. It is seen that the point cloud subsets sampled by RS and USNet lose their fine-grained information, resulting in the decline of segmentation accuracy. Compared with RS and USNet, the points sampled by the heuristic based sampling method FPS are uniform and can better retain the geometric structure of the original point cloud, so better results are achieved. However, the point cloud segmentation using the proposed UPSNet(v) or UPSNet for down sampling has best segmentation accuracy, and with the increase of the sampling rate *r*, the performance is improved more than other sampling methods. This shows that the proposed UPSNet(v) and UPSNet can retain more representative points from the original point cloud, so they can reconstruct more suitable point cloud subsets for subsequent point cloud segmentation and achieve better segmentation results. Moreover, generally speaking, the UPSNet with regional learning has better performance than the UPSNet(v) without regional learning module, which verifies the necessity and effectiveness of regional learning.

In addition to quantitative comparison, qualitative analysis is also done. The segmentation results with respect to different sampling methods are visualized, as shown in Figure 4 where the sampling size is 128. Figures 4(a) and 4(d) shows two original point clouds in which different colors represent different labels (components). Figures 4(b) and 4(e) show the sampled point clouds obtained with the sampling methods of RS, USNet, FPS, UPSNet(v) and UPSNet, respectively, where the color indicates the label of each point. Figures 4(c) and 4(f) show the segmentation results by putting the sampled point cloud into the PointNet segmentation network, that is, the different colors indicate the model's different component predicated by the PointNet. It is seen that the sampled results of RS, USNet and UPSNet(v) cannot well retain the shape of the input point clouds, as shown at the position of the aircraft nose. The sampled point clouds obtained by FPS are relatively uniform. However, the boundary at the connection of components may be blurred, leading to prediction errors. In contrast, the proposed UPSNet can achieve higher segmentation accuracy even at the connection of components on the premise of retaining the shape of each component.

Table 3

Point cloud segmentation accuracy with respect to different sampling methods (mIoU: mean Intersection over Union) unit: %

Point cloud sampling method	Sampling size (2048/*r*)				
	1024	512	256	128	64
RS	93.33	93.16	92.34	90.57	87.63
FPS	93.52	93.51	93.46	92.64	91.56
USNet	93.50	93.49	93.15	92.13	90.29
UPSNet(v)	**93.54**	93.52	93.49	92.85	91.76
UPSNet	93.52	**93.54**	**93.50**	**92.99**	**92.02**

3.4. Test with the Downstream Task of Point Cloud Reconstruction

Similar to image super-resolution reconstruction, point cloud reconstruction aims to reconstruct sparse samples into dense samples. When the reconstruction method is determined, the quality of the dense point clouds reconstructed from different sparse point clouds is different. Therefore, we can use the reconstruction task to evaluate the performance of different point cloud sampling methods. In this paper, the same evaluation index as in [18], that is, normalized reconstruction error (NRE), is used to normalize the reconstruction loss, which is defined by

$$N_{RE}(\boldsymbol{Q},\boldsymbol{P}) = \frac{CD(\boldsymbol{P},\mathrm{AE}(\boldsymbol{Q}))}{CD(\boldsymbol{P},\mathrm{AE}(\boldsymbol{P}))}, \tag{10}$$

where $CD(\cdot)$ is the Chamfer distance, $\boldsymbol{P}$ is the input point cloud, $\boldsymbol{Q}$ is the sampled point cloud, AE(·) denotes the reconstruction network and referring to [4, 18] an auto-encoder is adopted as the reconstruction network.

In order to better evaluate the performance of the sampling methods, we use four categories of models including aircraft, chair, lamp and table from ShapeNet dataset to form four datasets. Taking the aircraft as an example, the aircraft dataset is divided into the training set and test set at the ratio of 8:2. Then the training set is used to train the auto-encoder, where the number of points in the input and reconstructed point clouds is 2048. The trained auto-encoder is used to reconstruct the point clouds in the test set, and the Chamfer distance between the original point clouds in the test set and the reconstructed ones is taken as the denominator of Equation (10). The training set is also used to train the down sampling model so as to down sample the point clouds in the test set. Finally, the point clouds with 2048 points are reconstructed from the sampled test point clouds by the trained auto-encoder. The Chamfer distance between the original point cloud in the test set and the point cloud reconstructed with auto-encoder from the sampled test point cloud is regarded as the molecule of Equation (10). The final NRE results of experiments on ShapeNet dataset are shown in Table 4 where the best results are in bold. The comparative sampling methods are also tested on ModelNet40 dataset. The experimental procedures are similar to that on ShapeNet dataset except that the division of training set and test set follows the public dataset itself, but not the ratio of 8:2. The experimental results are shown in Table 5. Note that ModelNet40 dataset does not have segmentation labels, so we only test the UPSNet(v) instead of the UPSNet, because UPSNet(v) omits regional learning but only has global learning. From Tables 4 and 5, it is seen that when the number of input points is large (large sampling size), the difference between the reconstruction errors of all sampling methods is small. But when the number of points input for reconstruction is gradually reduced, the UPSNet(v) and UPSNet show their superiority because of the sampled representative points.

Table 4

Comparison of N_{RE} with respect to different sampling methods on ShapeNet dataset

Point cloud sampling method		Sampling size (2048/r)				
		1024	512	256	128	64
RS	aircraft	1.09	1.58	2.33	3.77	6.40
	chair	1.11	1.65	2.50	4.10	6.78
	lamp	1.08	1.21	1.54	2.15	3.29
	table	1.07	1.51	2.17	3.45	5.81
FPS	aircraft	1.00	1.21	1.55	2.26	3.71
	chair	1.02	1.20	1.68	2.54	4.35
	lamp	1.00	1.06	1.21	1.51	2.16
	table	1.01	1.13	1.47	2.13	3.79
USNet	aircraft	1.04	1.27	1.77	2.82	5.06
	chair	1.04	1.27	1.78	2.88	5.14
	lamp	1.02	1.10	1.29	1.67	2.36
	table	1.03	1.24	1.67	2.56	4.45
UPSNet(v)	aircraft	**1.01**	**1.20**	1.53	**2.23**	3.57
	chair	1.03	1.21	**1.62**	2.48	4.11
	lamp	**1.01**	**1.03**	1.17	1.46	2.03
	table	1.01	1.06	1.43	2.03	3.54
UPSNet	aircraft	1.01	1.21	**1.51**	2.24	**3.33**
	chair	**1.01**	**1.18**	1.63	**2.41**	**3.97**
	lamp	**1.00**	1.04	**1.15**	**1.43**	**1.98**
	table	**1.01**	**1.04**	**1.40**	**1.99**	**3.33**

Table 5

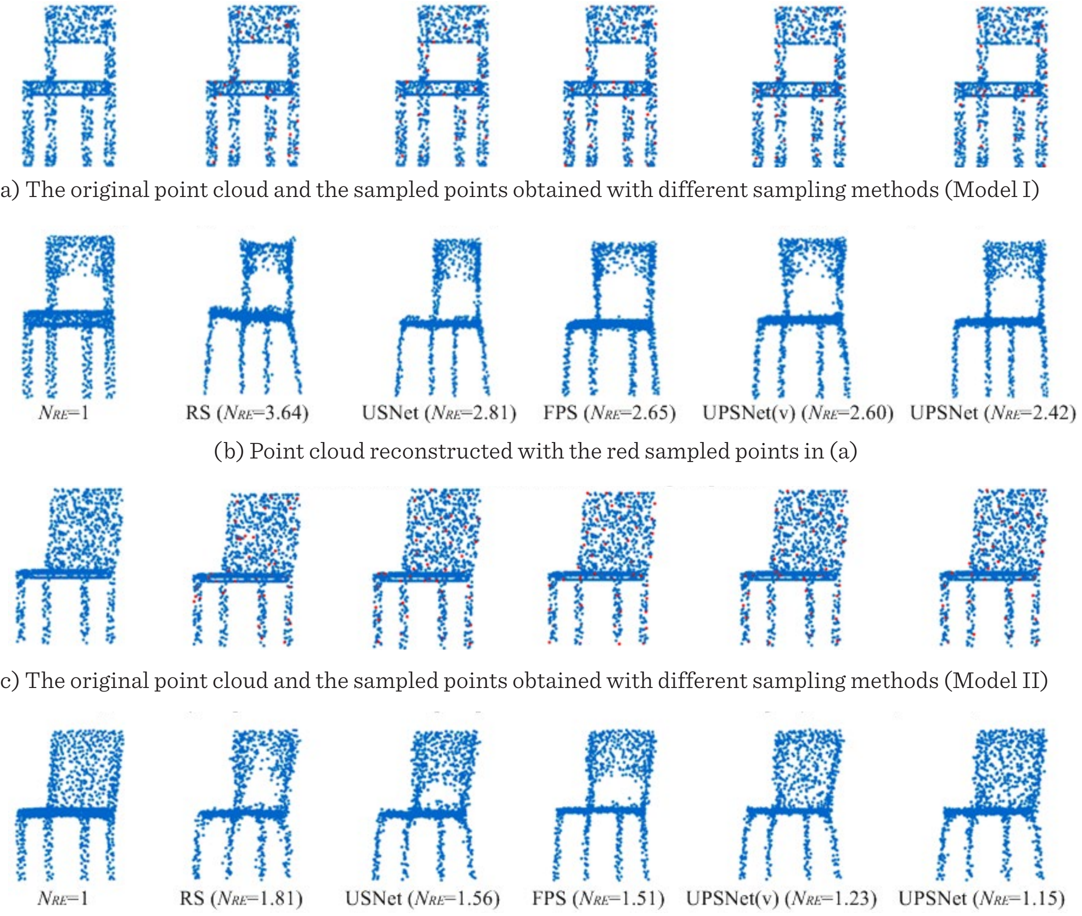
Comparison of N_{RE} with respect to different sampling methods on ModelNet40 dataset

Point cloud sampling method	Sampling size (1024/r)			
	512	256	128	64
RS	1.03	1.20	1.71	2.88
FPS	**1.02**	1.09	1.28	1.80
USNet	**1.02**	1.12	1.33	1.94
UPSNet(v)	**1.02**	**1.07**	**1.24**	**1.65**

We also visualize the reconstruction results of two models, as shown in Figure 5. Figures 5(a) and 5(c) show the original point clouds where the red points indicate the points sampled by different sampling methods which are the input of the reconstruction network. Figures 5(b) and 5(d) are the point cloud reconstructed with the red points in Figures 5(a) and 5(c), respectively. It is worth noting that due to the similarity of some models in the dataset, the point

Figure 5

Point cloud reconstruction with respect to different sampling methods



(a) The original point cloud and the sampled points obtained with different sampling methods (Model I)

(b) Point cloud reconstructed with the red sampled points in (a)

(c) The original point cloud and the sampled points obtained with different sampling methods (Model II)

(d) Point cloud reconstructed with the red sampled points in (c)

cloud sampled by FPS may be reconstructed with similar shape thus lose diversity. For example, both model I and model II in Figure 5 are chairs, but their backrest are different, one is a hollow backrest and the other is a solid backrest. However, both of their corresponding point clouds reconstructed by FPS are with hollow backrests. This shows that even the heuristic based sampling method may not be suitable for some subsequent tasks. In contrast, the sampled subsets of the input point cloud generated by UPSNet(v) and UPSNet are representative, which can make the network identify what kind of model it should reconstruct and maintain the diversity of the reconstructed models.

3.5. Test with the Downstream Task of 3D Model Retrieval

Model retrieval is to find similar model for a given model in the database. The general approach is to design a shape descriptor for each model and retrieve similar models by calculating the cosine distance between descriptors. Here, we use the output of the penultimate full connection layer of the above used classification network as the shape descriptor. Table 6 and Table 7 show the experimental results on ShapeNet and ModelNet40 datasets, respectively, the evaluation index is mAP (mean Average Precision) and the best results are in blod. For the same reason

Table 6

Comparison of retrieval performance with respect to different sampling methods on ShapeNet dataset (mAP: mean Average Precision) unit: %

Point cloud sampling method	Sampling size (2048/*r*)				
	1024	512	256	128	64
RS	89.4	88.8	87.9	86.9	82.1
FPS	89.3	89.1	89.1	87.8	85.6
USNet	89.3	89.3	89.0	88.2	86.1
UPSNet(v)	**89.4**	89.2	**89.1**	88.0	86.1
UPSNet	89.3	**89.4**	89.0	**88.4**	**86.6**

Table 7

Comparison of retrieval performance with respect to different sampling methods on ModelNet40 dataset (mAP: mean Average Precision) unit: %

Point cloud sampling method	Sampling size (1024/*r*)			
	512	256	128	64
RS	68.7	58.9	47.5	34.1
FPS	69.4	67.1	58.0	47.0
USNet	68.3	65.6	55.3	50.6
UPSNet(v)	**69.5**	**67.0**	**60.0**	**58.6**

as mentioned in subsection 4.4, we only test UPSNet(v) instead of the UPSNet on ModelNet40 dataset in this experiment. It is seen that the shape descriptor formed from the point clouds sampled by RS, FPS and USNet has poor retrieval performance when the number of points is small. The UPSNet(v) and UPSNet can generate more representative point cloud subsets, so the formed feature descriptor can retain the global features of the original point cloud, hence better retrieval performance can be achieved.

4. Conclusion

This paper has proposed an universal point cloud sampling network (namely UPSNet) which is independent of the downstream task, in other words, the proposed UPSNet is not oriented to specific downstream tasks, so it has wide applicability. By learning the mutual information between the input variables, the UPSNet can distinguish the important information and irrelevant information in the input variables, and obtain a group of variational importance probability. Then, the high dimensional space embedding of each region is multiplied by the variational importance probability to obtain the representative global features of each region. Finally, the global features and the high dimensional space embedding of each region are cascaded for learning and reconstruction, and the sampled point cloud is obtained. The experimental results show that the proposed UPSNet is superior to the commonly used sampling methods independent of downstream tasks in the tasks of point cloud classification, segmentation, reconstruction and retrieval.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61871247, and it was also sponsored by the K.C. Wong Magna Fund of Ningbo University.

References

1. Aoki, Y., Goforth, H., Srivatsan, R. A., Lucey, S. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, June 15-20 2019, 7156-7165. https://doi.org/10.1109/CVPR.2019.00733
2. Benhabiles, H., Aubreton, O., Barki, H., Tabia, H. Fast Simplification with Sharp Feature Preserving for 3D Point Clouds. International Symposium on Programming and Systems, Algiers, Algeria, Apr. 22-24 2013, 47-52. https://doi.org/10.1109/ISPS.2013.6581492
3. Dinesh, C., Cheung, G., Wang, F., Bajić, I. V. Sampling of 3D Point Cloud via Gershgorin Disc Alignment. IEEE International Conference on Image Processing (ICIP), Abu Dhabi, UAE, Oct 25-28 2020, 2736-2740. https://doi.org/10.1109/ICIP40778.2020.9190731
4. Dovrat, O., Lang, I., Avidan, S. Learning to Sample. IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, USA, June 15-20 2019, 2760-2769. https://doi.org/10.1109/CVPR.2019.00287
5. Fan, H., Su, H., Guibas, L. A Point Set Generation Network for 3D Object Reconstruction from a Single Ima-

ge. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, July 21-26 2017, 2463-2471. https://doi.org/10.1109/CVPR.2017.264

6. Fernandes, D., Silva, A., Nevoa, R., Simoes, C., Gonzalez, D., Guevara, M., Novais, P., Monteiro, J., Melo-Pinto, P. Point-cloud Based 3D Object Detection and Classification Methods for Self-driving Applications: A Survey and Taxonomy. Information Fusion, 2021, 68, 161-191. https://doi.org/10.1016/j.inffus.2020.11.002

7. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-scale Point Clouds. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19 2020, 11105-11114. https://doi.org/10.1109/CVPR42600.2020.01112

8. Kulikajevas, A., Maskeliunas, R., Damasevicius, R., Scherer, R. HumanNet--A Two-tiered Deep Neural Network Architecture for Self-occluding Humanoid Pose Reconstruction. Sensors, 2021, 21, 3945. https://doi.org/10.3390/s21123945

9. Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O. Point Pillars: Fast Encoders for Object Detection from Point Clouds, IEEE/CVF Conference on Computer Vision and Pattern Recognition Long Beach, USA, June 15-20 2019, 12689-12697. https://doi.org/10.1109/CVPR.2019.01298

10. Leal, N., Leal, E., German, S. A Linear Programming Approach for 3D Point Cloud Simplification. IAENG International Journal of Computer Science, 2017, 44(1), 60-67. http://www.iaeng.org/IJCS/issues_v44/issue_1/IJCS_44_1_08.pdf

11. Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M. A., Cao, D., Li, J. Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(8), 3412-3432. https://doi.org/10.1109/TNNLS.2020.3015992

12. Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M. A. Multi-scale Point-wise Convolutional Neural Networks for 3D Object Segmentation from Lidar Point Clouds in Large-scale Environments. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(2), 821-836. https://doi.org/10.1109/TITS.2019.2961060 https://doi.org/10.1109/TITS.2019.2961060

13. Nezhadarya, E., Taghavi, E., Razani, R., Liu, B., Luo, J. Adaptive Hierarchical Down-sampling for Point Cloud Classification. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19 2020, 12953-12961. https://doi.org/10.1109/CVPR42600.2020.01297

14. Nocerino, E., Stathopoulou, E. K., Rigon, S., Remondino, F. Surface Reconstruction Assessment in Photogrammetric Applications. Sensors, 2020, 20, 5863. https://doi.org/10.3390/s20205863

15. Qi, C., Su, H., Mo, K., Guibas, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, July 21-26 2017, 77-85. https://doi.org/10.1109/CVPR.2017.16

16. Qi, C., Yi, L., Su, H., Guibas, L. J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in A Metric Space. Neural Information Processing Systems (NIPS), California USA, Dec. 4 - 9, 2017, 5099-5108. https://dl.acm.org/doi/pdf/10.5555/3295222.3295263

17. Qi, J., Hu, W., Guo, Z. Feature Preserving and Uniformity-controllable Point Cloud Simplification on Graph. IEEE International Conference on Multimedia and Expo, Shanghai, China, July 08-12 2019. https://doi.org/10.1109/ICME.2019.00057

18. Qian, Y., Hou, J., Zeng, Y., Zhang, Q., Kwong, S., He, Y. MOPS-Net: A Matrix Optimization-driven Network for Task-oriented 3D Point Cloud Downsampling. 2020, arXiv:2005.00383v2[cs.cv]. https://doi.org/10.48550/arXiv.2005.00383

19. Rubner, Y., Tomasi, C. Guibas, L. J. The Earth Mover‘s Distance as A Metric for Image Retrieval. International Journal of Computer Vision, 2000, 40(2), 99-121. https://doi.org/10.1023/A:1026543900054 https://doi.org/10.1023/A:1026543900054

20. Song, H., Feng, H. A Progressive Point Cloud Simplification Algorithm with Preserved Sharp Edge Data. The International Journal of Advanced Manufacturing Technology, 2009, 45(5-6), 583-592. https://doi.org/10.1007/s00170-009-1980-4

21. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M., Kautz, J. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, June 18-23 2018, 2530-2539. https://doi.org/10.1109/CVPR.2018.00268

22. Sun, Y., Wang, Y., Liu, Z., Siegel, J. E., Sarma, S. E. PointGrow: Autoregressively Learned Point Cloud Generation with Self-attention. IEEE Winter Conference on Applications of Computer Vision, Snowmass, USA, March 1-5 2020, 61-70. https://doi.org/10.1109/WACV45572.2020.9093430

23. Taghanaki, S. A., Hassani, K., Jayaraman, P. K., Khasahmadi, A. H., Custis, T. PointMask: Towards Interpre-

table and Bias-resilient Point Cloud Processing. 2020, arXiv:2007.04525v1[cs.cv]. https://doi.org/10.48550/arXiv.2007.04525

24. Uy, M., Angelina, M., Lee, G. PointNetVLAD: Deep Point Cloud based Retrieval for Large-scale Place Recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, June 18-23 2018, 4470-4479. https://doi.org/10.1109/CVPR.2018.00470

25. Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics, 2019, 38(5), Art. no.146. https://doi.org/10.1145/3326362

26. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J. 3D ShapeNets: A Deep Representation for Volumetric Shapes. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7-12 2015, 1912-1920. https://doi.org/10.1109/CVPR.2015.7298801

27. Xu, T., An, D., Jia, Y., Yue, Y. A Review: Point Cloud-based 3D Human Joints Estimation. Sensors, 2021, 21(5), 1684. https://doi.org/10.3390/s21051684

28. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S. PointASNL: Robust Point Clouds Processing Using Nonlocal Neural Networks with Adaptive Sampling. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19 2020, 5588-5597. https://doi.org/10.1109/CVPR42600.2020.00563

29. Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., Tian, Q. Modeling Point Clouds with Self-Attention and Gumbel Subset Sampling. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, June 15-20 2019, 4470-4479. https://doi.org/10.1109/CVPR42600.2020.01112

30. Yang, X., Matsuyama, K., Konno, K., Tokuyama, Y. Feature-preserving Simplication of Point Cloud by Using Clustering Approach based on Mean Curvature. The Society for Art and Science, 2015, 14(4), 117-128.

31. Yi, L., Kim, V., Ceylan, D., Shen, I., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L. A Scalable Active Framework for Region Annotation in 3D Shape Collections. ACM Transactions on Graphics, 2016, 35(6), Art. no. 210. https://doi.org/10.1145/2980179.2980238 https://doi.org/10.1145/2980179.2980238

32. Ying, X., Xin, S., Sun, Q., He, Y. An Intrinsic Algorithm for Parallel Poisson Disk Sampling on Arbitrary Surfaces. IEEE Transactions on Visualization & Computer, 2013, 19(9), 1425-1437. https://doi.org/10.1109/TVCG.2013.63

33. Yu, Z., Wong, H., Peng, H., Ma, Q. ASM: An Adaptive Simplification Method for 3D Point-based Models. Computer Aided Design, 2010, 42(7), 598-612. https://doi.org/10.1016/j.cad.2010.03.003

34. Yuan, H., Zhang, D., Wang, W., Li, Y. A Sampling-based 3D Point Cloud Compression Algorithm for Immersive Communication. Mobile Networks and Applications, 2020, 25(5), 1863-1872. https://doi.org/10.1007/s11036-020-01570-y

35. Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, „PF-Net: Point Fractal Network for 3D Point Cloud Completion. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 13-19 2020, 7662-7670. https://doi.org/10.1109/CVPR42600.2020.00768

36. Zhang, Y., Liu, Z., Liu, T., Peng, B., Li, X. RealPoint3D: An Efficient Generation Network for 3D Object Reconstruction from a Single Image. IEEE Access, 2019, 7, 57539-57549. https://doi.org/10.1109/ACCESS.2019.2914150

37. Zhang, Y., Huo, K., Liu, Z., Zang, Y., Liu, Y., Li, X., Zhang, Q., Wang, C. PGNet: A Part-based Generative Network for 3D Object Reconstruction. Knowledge-Based Systems, 2020, 194, Art. no.105574. https://doi.org/10.1016/j.knosys.2020.105574

38. Zhang, W., Su, S., Wang, B., Hong, Q., Sun, L. Local k-NNs Pattern in Omnidirection Graph Convolution Neural Network for 3D Point Clouds. Neurocomputing, 2020, 413, 487-498. https://doi.org/10.1016/j.neucom.2020.06.095

39. Zhu, L., Wang, B., Tian, G., Wang, W., Li, C., Towards Point Cloud Completion: Point Rank Sampling and Cross-cascade Graph CNN. Neurocomputing, 2021, 461, 1-16. https://doi.org/10.1016/j.neucom.2021.07.035