CAS-NET: CASCADE ATTENTION-BASED SAMPLING NEURAL NETWORK FOR POINT CLOUD SIMPLIFICATION

Chen Chen[1], Hui Yuan[2](corresponding author), Hao Liu[3], Junhui Hou[4], and Raouf Hamzaoui[5]

[1,2]School of Control Science and Engineering, Shandong University, Jinan, 250061, China
[3]School of Computer and Control Engineering, Yantai University, Yantai, 264005, China
[4]Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, China
[5] School of Engineering and Sustainable Development, De Montfort University, Leicester, UK.
chenc_nj@mail.sdu.edu.cn, huiyuan@sdu.edu.cn, liuhaoxb@gmail.com,
jh.hou@cityu.edu.hk, rhamzaoui@dmu.ac.uk

ABSTRACT

Point cloud sampling can reduce storage requirements and computation costs for various vision tasks. Traditional sampling methods, such as farthest point sampling, are not geared towards downstream tasks and may fail on such tasks. In this paper, we propose a cascade attention-based sampling network (CAS-Net), which is end-to-end trainable. Specifically, we propose an attention-based sampling module (ASM) to capture the semantic features and preserve the geometry of the original point cloud. Experimental results on the ModelNet40 dataset show that CAS-Net outperforms state-of-the-art methods in a sampling-based point cloud classification task, while preserving the geometric structure of the sampled point cloud.

Index Terms— Point Clouds, Attention-based Sampling

1. INTRODUCTION

With the rapid development of modern media technology, point clouds have gained increasing popularity for many applications. Unlike other 3D representations, the 3D point cloud representation records both the geometry coordinates and the attributes of points located on the surface of the object. Typically, a point cloud consists of thousands of points. Due to hardware limitations, processing so many points is not desirable. Thus, point cloud down-sampling is an essential step before subsequent tasks. Farthest point sampling [1], which is one of the traditional down-sampling approaches, only considers the Euclidean distance between points. Although it can cover the whole point cloud uniformly through iterations, it does not consider the semantical representation of the point cloud, which may limit the performance of any subsequent downstream tasks. Building on the success of deep learning-based methods for point cloud recognition [1] [2] [3] [4] and generation tasks [5] [6] [7], researchers developed deep learning-based down-sampling methods to simplify the point cloud as well as preserve the latent semantic features. Although these methods can retain semantic information for the downstream tasks, the performance of the downstream tasks declines significantly when the sampling ratio is high. Besides, the geometric structure of the sampled point cloud tends to be quite different from that of the original one. For example, sampling may be concentrated around some important points, neglecting other parts of the original point cloud. If only the semantic features are considered, the geometry structure cannot be preserved well and thus the quality of high-resolution reconstruction will drop severely. Conversely, if we only concentrate on the geometry structure, the accuracy of identification cannot be guaranteed. Therefore, one of the main problems in a sampling task is to balance the semantic feature and geometry structure. Inspired by the work of [8] [9] [10], we introduce an attention mechanism to enhance the important features while neglecting the redundant ones. We also combine the output of different layers to achieve a balance between semantic features and geometry features. Our main contributions are as follows:

- An end-to-end trainable cascade attention-based sampling network (CAS-Net), in which the sampled points are a subset of the original point cloud.
- An attention-based sampling module (ASM) which is used to learn the attractive features for sampling.
- A joint loss function which takes into account both the semantic features and the geometry features.
- Experimental results show that the proposed network not only achieves state-of-the-art performance for a downstream classification task but also preserves the geometry structure of the original point clouds.

The remainder of this paper is organized as follows. In section II, we briefly review the related work on point cloud sampling. In Section III, the proposed neural network is presented in detail. Experimental results and conclusions are given in Section IV and V, respectively.

2. RELATED WORK

In general, point cloud down-sampling methods can be divided into traditional methods and learning-based methods. The learning-based methods are either generative methods or direct methods.

2.1. Traditional methods

Traditional methods are task agnostic. They use specific rules or statistical strategies to down-sample the input point cloud. The most popular traditional methods are random sampling (RS) [11], farthest point sampling (FPS) [1] and Poisson-disk sampling [12]. For RS, the probability of selecting a point from the original point cloud is even. This may result in the loss of geometry structure because of the uncertain sampling. FPS is a step-by-step process, in which the point farthest from the current sampling point is included in each step. FPS is based on the Euclidean distance and is able to provide a uniform result. But the computation complexity is high, especially for large-scale point clouds. Poisson-disk sampling requires that the distance between any two sampling points be greater than a given sampling radius. It can provide evenly distributed points with similar computational complexity to FPS.

2.2. Learning-based methods

Learning-based sampling methods can be divided into generative sampling methods and direct sampling methods depending on whether the sampled points belong to the original point clouds or not.

2.2.1. Generative sampling

Dovrat *et al.* [13] proposed a pioneering generative sampling network (S-Net) based on the structure of PointNet [2]. The down-sampled point cloud is generated through a fully connected layer and then the output points are matched to the nearest point in the input point set. However, the matching operation is not trainable. A later work, namely SampleNet [14], solved this problem by introducing a differential projecting operation during training. Though the sampled point cloud generated by this method can better approximate the original point cloud, it cannot preserve the local details well. Wang *et al.* [15] proposed the point sampling transformer network (PST-NET) which combines the idea of S-Net and the structure of transformer. This method is able to generate a noise-robust point cloud with the aid of the proposed transformer unit. Lin *et al.* [16] proposed a density-adaptive down-sampling network (DA-Net) which uses *k*-nearest neighbors to estimate the density of the points and a local adjustment module to improve noise immunity. One common drawback of these previous works is that the output point cloud is not strictly a subset of the input point cloud. Thus, these methods are unlike traditional sampling methods and are not always able to keep the shape of the output consistent with that of the original point cloud.

2.2.2. Direct sampling

Direct sampling methods select the points based on some specific rules or the embedded features. Nezhadarya *et al.* [17] proposed a critical points layer (CPL) and a weighed critical points layer (WCPL) which can sort the important points based on their contribution towards the point-wise feature map. Qian *et al.* [18] designed a matrix optimization-driven network (MOPS-Net) with a matrix optimization method, in which the problem of differentiability is solved by relaxing the binary restriction in the feature matrix. Yang *et al.* [19] proposed a gumbel subset sampling (GSS) method which solves the trainable problem by adding gumbel noise in the extracted feature. They also proposed a channel shuffle operation to improve the efficiency of feature interaction. Sun *et al.* [20] introduced a straight sampling network with a gradient estimation strategy to train a hard sampling network.

In this study, we propose an attention-based sampling module to capture the most attractive information in the embedded features, and thus effectively sample points from the original point cloud by a learned sampling matrix. The proposed network is also a direct sampling method which can preserve the geometry structure of the input point cloud well.

3. PROPOSED METHOD

The goal of the proposed down-sampling method is to find an optimal solution to a specific downstream task (e.g, classification) while maintaining the geometry structure of the point cloud. The architecture of the proposed network is given in Fig. 1. Given the input point cloud $\mathbf{P}_{in}$ and a network for a specified task, the proposed network first uses a feature embedding module to capture the local and global features. Based on the feature map, an attention-based sampling module is proposed to enhance the attractive features and obtain a sampling matrix $\mathbf{S}$. Then, the down-sampled point cloud $\mathbf{P}_{sp}$ is obtained by multiplying $\mathbf{S}$ and $\mathbf{P}_{in}$. Finally, $\mathbf{P}_{sp}$ is fed to the task network for the specific downstream task. The whole procedure can be trained by an end-to-end scheme with a composite loss function.

3.1. Feature embedding module

As shown in Fig. 2, the input of the feature embedding module is an unordered point cloud with *n* points. First, we encode the unordered points by mapping the input to a vector in a hyper-space. To get point-wise features from the input point cloud, we adopt the grouping layer [1] because of its high efficiency. For a given point $\mathbf{p}$ in the input point set $\mathbf{P}_{in}$,

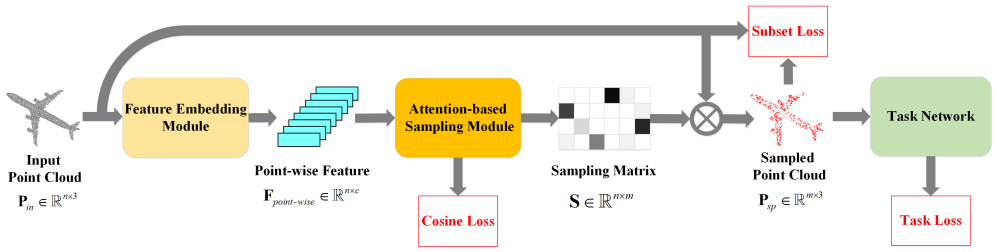

Fig. 1. Architecture of CAS-Net.

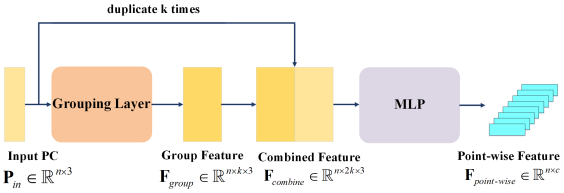

Fig. 2. Feature embedding module.

the grouping operation is defined as

$$Group(\mathbf{p}) = \{\mathbf{p}_1 - \mathbf{p}, \mathbf{p}_2 - \mathbf{p}, , ..., \mathbf{p}_k - \mathbf{p}\}, \quad (1)$$

where $\mathbf{p}_i$, i=1,...,k represent k neighbor points of $\mathbf{p}$. The output of the grouping layer can be written as

$$\mathbf{F}_{group} = Group\left(\mathbf{P}_{in}\right). \quad (2)$$

To better preserve the global geometry information, we then duplicate the input point cloud k times and combine the output with the group feature

$$\mathbf{F}_{combine} = concat\left(\underbrace{\mathbf{P}_{in}, \mathbf{P}_{in}, ..., \mathbf{P}_{in}}_{k\ times}, \mathbf{F}_{group}\right). \quad (3)$$

Finally, we use a multi-layer perceptron (MLP) $\sigma(\cdot)$ to map the combined feature to $\mathbf{F}_{point-wise}$:

$$\mathbf{F}_{point-wise} = \sigma(\mathbf{F}_{combine}). \quad (4)$$

3.2. Attention-based sampling module

As we aim at selecting points from existing points, we propose an attention mechanism that identifies and captures the most attractive points during training. Our self-attention (SA) mechanism can be formulated as

$$(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{F}_{in} \cdot (\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v), \quad (5)$$

$$\mathbf{F}_{sa} = softmax\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}} \cdot \mathbf{V}\right), \quad (6)$$

where $\mathbf{F}_{in}$ represents the input features, $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are the shared learnable linear transformations with the same dimension and d_k is the dimension of $\mathbf{W}_k$. The self-attention layer can be written as

$$\mathbf{F}_{out} = \gamma\left(\mathbf{F}_{sa}\right) + \mathbf{F}_{in}, \quad (7)$$

where $\gamma(\cdot)$ represents an MLP operation. However, the self-attention layer cannot handle the problem of information loss when the network is deeper [21]. By taking the difference between the attention features and the input features into account, we use offset attention (OA) [22] to modify the feature

$$\mathbf{F}_{out} = OA\left(\mathbf{F}_{in}\right) = \gamma\left(\mathbf{F}_{in} - \mathbf{F}_{sa}\right) + \mathbf{F}_{in}. \quad (8)$$

To preserve the geometric features together with the semantic features, information fusion is needed between layers. Therefore, we design an attention-based sampling module (ASM) consisting of three skip connected OA layers (Fig. 3). The output of each layer is then concatenated along the feature dimension:

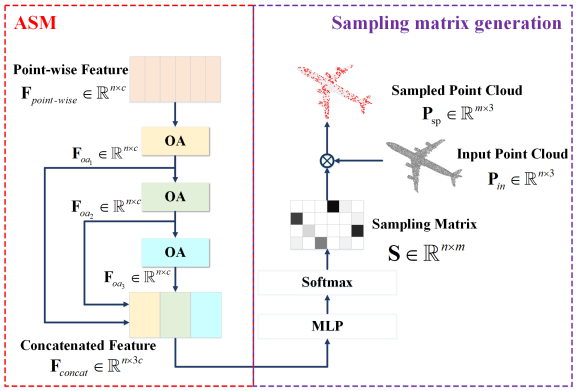

Fig. 3. Overview of proposed ASM.

$$\begin{aligned}&(\mathbf{F}_{oa_1}, \mathbf{F}_{oa_2}, \mathbf{F}_{oa_3}) = \\ &(OA(\mathbf{F}_{point-wise}), OA(\mathbf{F}_{oa_1}), OA(\mathbf{F}_{oa_2}))\,,\end{aligned} \quad (9)$$

$$\mathbf{F}_{concat} = concat(\mathbf{F}_{oa_1}, \mathbf{F}_{oa_2}, \mathbf{F}_{oa_3}), \quad (10)$$

where $\mathbf{F}_{oa_i}$ represents the output feature of the i-th OA layer and $\mathbf{F}_{concat}$ is the concatenated feature.

3.3. Sampling matrix generation

The concatenated feature $\mathbf{F}_{concat}$ is then used to predict a sampling matrix via MLP and the softmax function:

$$\tilde{\mathbf{S}} = softmax\left(\rho\left(\mathbf{F}_{concat}\right)\right), \quad (11)$$

where $\rho(\cdot)$ denotes the MLP operation and $\tilde{\mathbf{S}}$ is the soft sampling matrix which is learnable. For simplicity, to transform $\tilde{\mathbf{S}}$ into a binary matrix $\mathbf{S}$, we set the largest element of $\tilde{\mathbf{S}}$ in each row to one and the remaining elements to zero. Based on the binary matrix, the input point cloud can be down- sampled by a product operation. As this operation is not differentiable, we use the gradient of $\tilde{\mathbf{S}}$ to estimate the gradient of $\mathbf{S}$ in the backward propagation [20]. Accordingly, we have two methods for the sampling problem: attention-based hard sampling network (AHSN) and attention-based soft sampling network (ASSN). The only difference between them is that AHSN applies the strategy mentioned above, while ASSN uses the soft sampling matrix $\tilde{\mathbf{S}}$ directly. The down-sampled point cloud with ASSN is obtained by [23]

$$\mathbf{P}_{sp} = \tilde{\mathbf{S}}^T \mathbf{P}_{in}, \quad (12)$$

while the down-sampled point cloud with AHSN is obtained by

$$\mathbf{P}_{sp} = \mathbf{S}^T \mathbf{P}_{in}. \quad (13)$$

3.4. Loss function

The proposed network is trained with a joint loss that consists of three parts:

$$L_{total} = L_{task}(\mathbf{P}_{sp}) + \alpha L_{subset}(\mathbf{P}_{in}, \mathbf{P}_{sp}) + \beta L_{cosine}(\tilde{\mathbf{S}}), \quad (14)$$

where $L_{task}(\cdot)$ aims to promote the network to learn an optimized down-sampled point set to the downstream task, $L_{subset}(\cdot)$ preserves the geometry structure of the down-sampled point cloud, $L_{cosine}(\cdot)$ is used to reduce the number of duplicate points, and α and β are weights used to balance the three parts. Specifically, $L_{subset}(\cdot)$ is used to ensure that $\mathbf{P}_{in}$ and $\mathbf{P}_{sp}$ are close to each other as follows [13][14]:

$$L_{subset}(\mathbf{P}_{in}, \mathbf{P}_{sp}) = \frac{1}{|\mathbf{P}_{in}|} \sum_{x \in \mathbf{P}_{in}} \min_{y \in \mathbf{P}_{sp}} ||x - y||_2^2 + \frac{1}{|\mathbf{P}_{sp}|} \sum_{y \in \mathbf{P}_{sp}} \min_{x \in \mathbf{P}_{in}} ||y - x||_2^2, \quad (15)$$

where the first term ensures that the points in $\mathbf{P}_{sp}$ are close to those in $\mathbf{P}_{in}$ while the second term ensures that the points in $\mathbf{P}_{sp}$ are distributed all over $\mathbf{P}_{in}$. However, the network may focus on a few important points, leading to an accumulation of duplicated points in the down-sampled point set [16]. To address this problem, we use the cosine loss $L_{cosine}(\cdot)$[20]

$$L_{cosine}(\tilde{\mathbf{S}}) = \sum_{i \neq j} |cos < \tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j > |, \quad (16)$$

where $\tilde{\mathbf{s}}_i$ and $\tilde{\mathbf{s}}_j$ are row vectors of $\tilde{\mathbf{S}}$.

4. EXPERIMENTAL RESULTS

We tested our network by using it to sample point clouds from the Modelnet40 dataset [24] and classifying the resulting point clouds with PointNet [2]. ModelNet40[24] contains 12,311 CAD models from 40 man-made object categories, which were split into 9,843 point clouds for training and 2,468 point clouds for testing. The input point clouds consisted of 1024 points. PointNet and the proposed down-sampling network were jointly trained. The task loss was the cross entropy between the predicted labels and the ground truth labels. We used random rotation and scaling of the input point clouds to improve the robustness of the network. The proposed method was implemented on the Pytorch platform with a batch size of 8. The total training epoch was set to 200 and the learning rate was set to 0.001. We set α = 1, β = 0.01, c = 64 and k = 32 during training. A computer with an Intel Core i7 7820X processor, an NVIDIA GeForce RTX2080 TI GPU, and 12GB memory was used to conduct the experiments.

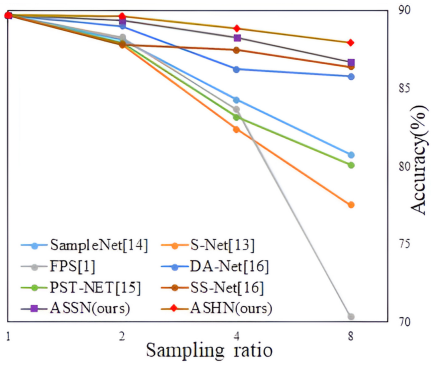

Fig. 4. Classification accuracy on Modelnet40.

4.1. Comparison with state-of-the-art methods

Since our model can adapt to any sampling ratio (the ratio between the number of points *n* in the input point cloud and the number of points *m* in the down-sampled point cloud), we tested the classification accuracy at various sampling ratios. To assess the performance, we also compared the proposed ASSN and AHSN with five state-of-the-art methods: S-Net [13], SampleNet [14], PST-NET [15], DA-Net [16], and SS-Net [20]. In addition, we also used FPS [1] as a benchmark. Except for SS-Net[20], all other methods use the same classification network (PointNet). To ensure a fair comparison between all methods, we modified the network structure of SS-Net by removing its fully connected layers and sending the output point cloud of the straight sampling (SS) module to PointNet for classification. Table 1 gives the classification accuracy of the methods for various sampling ratios. Fig. 4 illustrates the results for sampling ratios 2, 4 and 8. We can see that the classification accuracy of all the methods decreased with the increase of the sampling ratio. When the sampling ratio was 2, FPS achieved comparable performance with the other methods. With the increase of sampling ratio, the performance of FPS decreased quickly. Similar results were observed for S-Net [13], SampleNet [14], and PST-NET [15]. For DA-Net [16] and SS-Net [20], the classification accuracy varied only slightly with the increase of the sampling ratio. In particular, the classification accuracy was still larger than 85% for a sampling ratio of 8. The performance of the proposed methods always achieved the best (with AHSN) and the second best (with ASSN) results, indicating the efficiency of the proposed network.

Fig. 5 visualizes the point clouds generated by S-Net [13], SampleNet [14], FPS[1] and AHSN. We can see that AHSN can better preserve the geometry of the point cloud

Table 1. Classification accuracy in percentage(%)

m	Sampling ratio	FPS[1]	S-Net[13]	Sample Net[14]	PST-NET[15]	DA-Net[16]	SS-Net[20]	ASSN (proposed)	AHSN (proposed)
512	2	88.30	87.80	88.16	87.94	89.01	87.84	89.38	**89.62**
256	4	83.64	82.38	84.27	83.15	86.24	87.47	88.29	**88.86**
128	8	70.34	77.53	80.75	80.11	85.67	86.39	86.70	**87.92**

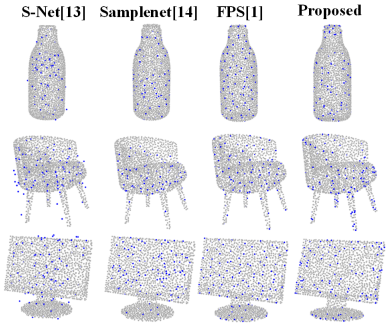

Fig. 5. Comparison between AHSN and state-of-the-art sampling methods (sampling ratio = 16). The sampled points are shown in blue.

than the other learning-based methods. Although FPS provided a more uniform sampling, its classification performance was not satisfactory when the sampling ratio was high.

We also compared the time complexity of AHSN with S-Net [13], SampleNet [14] and FPS [1] on the test dataset, as shown in Table 2. We can see that S-Net had the lowest time complexity, followed by SampleNet, FPS, and our method.

Table 2. Time complexity

Model	Time(s)
S-Net [13]	**10.95**
SampleNet [14]	12.85
FPS[1]	23.60
AHSN	40.48

4.2. Ablation study

To verify the effectiveness of the OA-based attention feature extraction, we replaced the OA module by the SA module. The results in Table 3 show that the proposed method improved the effectiveness of the extracted features, which increased the classification accuracy of the sub-sampled point clouds. We further verified the effectiveness of ASM by visualizing the output feature map of each layer (Fig. 6). We can see that the initial point-wise feature distributed chaotically. With the aid of the OA modules, the features corresponding to each point gradually changed in each OA module. Specifically, the features of the attractive points were strengthened, while those of the other points were suppressed.

We also checked the effectiveness of the proposed loss functions (Table 4). The results show that the subset loss significantly improved the accuracy of both ASSN and AHSN. The cosine loss was helpful to AHSN because it reduced the number of duplicate points in the output point set, thus preserving more geometric structure for the classification task.

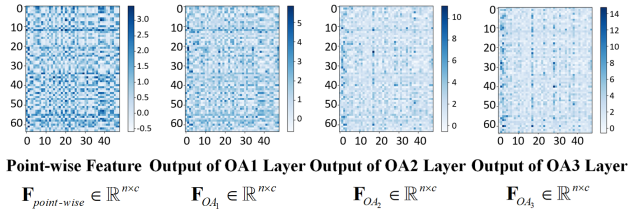

Fig. 6. Visualization of the feature map with 48 points from the test point cloud.

Table 3. Comparison between OA and SA

Models	Accuracy(%)
ASSN(OA)	89.38
AHSN(OA)	**89.62**
ASSN(SA)	88.31
AHSN(SA)	88.29

Table 4. Effectiveness of the loss functions

Model	Cosine Loss	Subset Loss	Accuracy(%)
ASSN	✓		88.16
		✓	89.38
	✓	✓	88.97
AHSN	✓		89.10
		✓	88.81
	✓	✓	**89.62**

When both the cosine loss and subset loss were used, the performance of AHSN was the best. However, the performance of ASSN decreased slightly.

5. CONCLUSION

We proposed a point cloud sampling network based on a learned sampling matrix. We introduced an attention-based sampling module to enhance the extracted features and maintain the geometry structure of the input, and then proposed two methods (hard sampling and soft sampling) to sample the points directly from the input point cloud. Experimental results demonstrated that the proposed network outperformed existing methods in point cloud classification. Moreover, the proposed network preserved the geometry structure better than other learning-based methods. In the future, we will focus on reducing the computation complexity and test our network on more downstream tasks.

6. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62222110 and 62172259, the Taishan Scholar Project of Shandong Province (tsqn202103001), the Natural Science Foundation of Shandong Province under Grant ZR2022ZD38, the Central Guidance Fund for Local Science and Technology Development of Shandong Province, under Grant YDZX2021002, and the OPPO Research Fund.

7. REFERENCES

[1] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[2] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[3] Wenxuan Wu, Zhongang Qi, and Li Fuxin, "PointConv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[7] Hao Liu, Hui Yuan, Junhui Hou, Raouf Hamzaoui, and Wei Gao, "PUFA-GAN: A frequency-aware generative adversarial network for 3d point cloud upsampling," *IEEE Transactions on Image Processing*, vol. 31, pp. 7389–7402, 2022.

[8] Yufeng Yang, Yixiao Ma, Jing Zhang, Xin Gao, and Min Xu, "AttPNet: Attention-based deep neural network for 3d point set analysis," *Sensors*, vol. 20, no. 19, 2020.

[9] Yang Ye, Xiulong Yang, and Shihao Ji, "APSNet: Attention based point cloud sampling," in *Proceedings of the Conference on British Machine Vision Conference (BMVC)*, 2022.

[10] Yakun Yang, Anhong Wang, Donghan Bu, Zewen Feng, and Jie Liang, "AS-Net: An attention-aware downsampling network for point clouds oriented to classification tasks," *Journal of Visual Communication and Image Representation*, vol. 89, pp. 103639, 2022.

[11] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] Xiang Ying, Shi-Qing Xin, Qian Sun, and Ying He, "An intrinsic algorithm for parallel poisson disk sampling on arbitrary surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 9, pp. 1425–1437, 2013.

[13] Oren Dovrat, Itai Lang, and Shai Avidan, "Learning to sample," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Itai Lang, Asaf Manor, and Shai Avidan, "SampleNet: Differentiable point cloud sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] Xu Wang, Yi Jin, Yigang Cen, Congyan Lang, and Yidong Li, "PST-NET: Point cloud sampling via point-based transformer," in *Proceedings of the International Conference on Image and Graphics*, 2021, pp. 57–69.

[16] Yanan Lin, Yan Huang, Shihao Zhou, Mengxi Jiang, Tianlong Wang, and Yunqi Lei, "DA-Net: Density-adaptive downsampling network for point cloud classification via end-to-end learning," in *Proceedings of the Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, 2021, pp. 13–18.

[17] Ehsan Nezhadarya, Ehsan Taghavi, Ryan Razani, Bingbing Liu, and Jun Luo, "Adaptive hierarchical down-sampling for point cloud classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Yue Qian, Junhui Hou, Qijian Zhang, Yiming Zeng, Sam Kwong, and Ying He, "MOPS-Net: A matrix optimization-driven network for task-oriented 3d point cloud downsampling," *arXiv preprint arXiv:2005.00383*, 2020.

[19] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3323–3332.

[20] Ran Sun, Gaojie Chen, Jie Ma, and Pei An, "Straight sampling network for point cloud learning," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3088–3092.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[23] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in neural information processing systems*, vol. 31, 2018.

[24] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.