This is the Pre-Published Version.
The following publication W. Zeng, Y. Li, R. Chen, R. Xiang, Y. Wang and J. Gu, "MGE-Net: Task-oriented Point Cloud Sampling based on Multi-scale Geometry Estimation," 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 2024, pp. 2822-2827 is available at https://doi.org/10.1109/CSCWD61410.2024.10580588.

MGE-Net: Task-oriented Point Cloud Sampling based on Multi-scale Geometry Estimation

Weiliang Zeng
Xi'an Jiaotong-Liverpool University
Suzhou, China
Weiliang.Zeng22@student.xjtlu.edu.cn

Yushi Li*
Department of Intelligent Science
Xi'an Jiaotong-Liverpool University
Suzhou, China
Yushi.Li@xjtlu.edu.cn

Rong Chen*
College of Information Science and Technology
Dalian Maritime University
Dalian, China
rchen@dlmu.edu.cn

Rong Xiang
Department of Computing
The Hong Kong Polytechnic University
Hong Kong, China
rongxiang@polyu.edu.hk

Yunzhe Wang
Suzhou University of Science and Technology
Suzhou, China
yunzhe1991@mail.usts.edu.cn

Jinghang Gu
Chinese and Bilingual Studies
The Hong Kong Polytechnic University
Hong Kong, China
gujinghangnlp@gmail.com

***Abstract*—A large number of collaborative manufacturing tasks are directly performed on point clouds. With the growing size of point clouds, the computational demands of these tasks also increase. One possible solution is to sample the point clouds. The most commonly used sampling method is farthest point sampling, but it does not consider downstream tasks, often leading to sampling non-informative points for the tasks. With the development of neural networks, various methods have been proposed to sample point clouds in a task-oriented learning manner. However, most methods are based on generation rather than selecting a subset of point clouds. In this work, we propose a novel adaptive keypoint sampling method, called MGE-Net, that combines neural network-based learning with direct point selection based on multi-scale geometry estimation. In addition, we design a feature extraction module based on multi-scale attention graph convolution to provide accurate information for subsequent keypoint detection. Relying on the contribution of point clouds to the task, our framework aims to sample a subset of point clouds specifically optimized for downstream tasks. Both qualitative and quantitative experimental results demonstrate that our sampling method exhibits superior performance in common point cloud classification and segmentation tasks.**

***Index Terms*—point clouds, task-oriented sampling, collaborative manufacturing, classification, segmentation**

I. Introduction

Point clouds have become more popular and significant in the domains of intelligent robots, automated driving, and virtual reality in recent years. However, dealing with many point clouds also becomes a challenge. For many applications, reducing the number of points can be beneficial in many ways. For example, reducing the number of points can reduce power consumption, computational cost, and communication load.

Point cloud sampling is selecting representative point cloud subsets from 3D data to reduce data redundancy and improve data processing efficiency. Although there are many heuristic-based sampling methods, such as random sampling (RS) and farthest point sampling (FPS) [8], these methods are task-independent and do not consider the needs of subsequent processing tasks when selecting sampling points. Therefore, they may choose points without information about down-stream tasks, affecting processing performance. Recently, task-oriented sampling networks have received attention from researchers. S-NET [9] and SampleNet [10] are representatives of this type of network, which use a simple multi-layer perceptron (MLP) to resample the original point cloud and generate a new set of point clouds that meet the required size. The advantage of these task-oriented sampling networks is that they can learn better sampling strategies to generate the minimum number of sampling points to optimize the performance of downstream tasks. However, they generate a new point cloud rather than directly selecting the desired points from the original point cloud. This indirect method may add additional computational overhead for application scenarios that require direct point selection, such as classification, segmentation, and 3D reconstruction.

In this paper, we propose a task-oriented 3D keypoint detection network that aims to optimize the performance of downstream tasks by sampling more key points. Our method not only selects a subset from the original point cloud but is also adaptive and learns to downsample points during training to ensure that critical points are not lost. Firstly, the network improves the original feature extraction method using edge convolutional layers with different scales for local feature extraction. By adding a self-attention layer, the network can better focus on important information and obtain features with richer local details. This improved feature extraction method can better capture local details and provide more accurate information for subsequent keypoint detection. Second, based on the extracted local features of the point cloud, the network selects key points that contain essential information through the channel and curvature sampling layers. These key points are selected by further analysis of the local features and play

* correspondence author

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

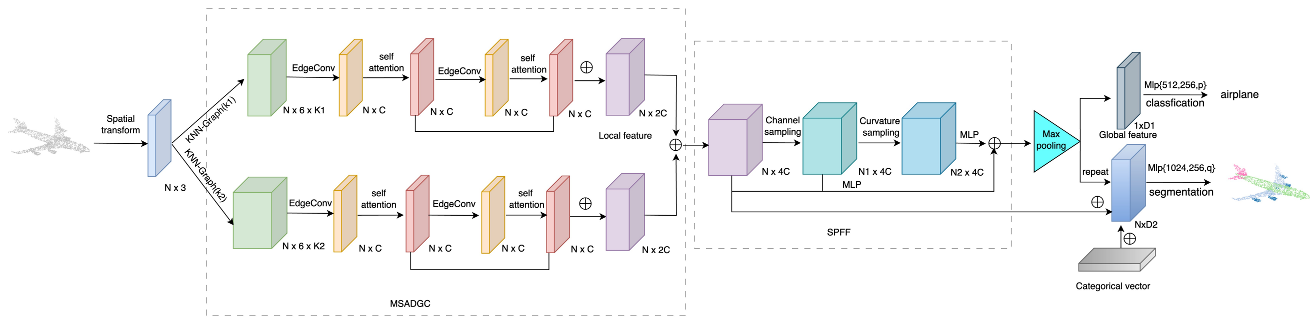

Fig. 1. Overall architecture of MGE-Net

an important role in performance optimization for downstream tasks. By fusing the features of the down-sampled points, the network can maintain high performance and accuracy when processing large-scale point cloud data. Our main contributions are summarised as follows:

- We design a feature extraction module based on multi-scale attention graph convolution to provide accurate information for subsequent key point detection.
- We propose an adaptive keypoint sampling method that combines neural network-based learning and direct point selection based on multi-scale geometry estimation.
- Good qualitative and quantitative results have been achieved on the common point cloud benchmark, demonstrating the effectiveness of the proposed sampling method.

II. RELATED WORKS

A. Deep Learning on Point Clouds

Point clouds are sparse, disordered, and sensitive to locality, Recent methods focus on the direct processing of raw 3D points and can be broadly classified into three categories: the MLP-based methods [1], [2], the convolution-based methods [5]–[7] and the graph-based methods [3], [4]. PointNet [1] uses a shared multi-layer perception (MLP) and max-pooling layer to extract features. PointNet++ [2] uses hierarchical point feature aggregation to obtain global features. PointConv [23] and KPConv [6] propose point-wise convolution operators with which points are convoluted with neighbor points. The graph-based approach analyses the point cloud by using the graph structure. DGCNN [3] is designed with graph-based EdgeConv blocks to extract local features. Global information can be acquired by overlaying EdgeConv blocks. In self-supervised learning, HSGAN [24] and SG-GAN [25] employ GCN to infer complex 3D shapes from random noise. WalkFormer [26] employs a guided point walking method to achieve point cloud completion.

B. Point Cloud Sampling

Point cloud sampling has broadly used non-learning-based methods in recent decades. As a common method, farthest point sampling (FPS) [8] starts with a randomly selected point and iteratively selects the next point furthest from the selected point, resulting in comprehensive coverage of the input. However, the whole process is time-consuming. Random sampling (RS) entails the random selection of points and is characterized by the lowest computational burden. It is non-robust to noisy points and often suffers from information loss. Point cloud sampling has improved recently thanks to task-oriented learning-based sampling algorithms. Recently, S-NET [9] and SampleNet [10] have shown that sampling networks can learn better sampling strategies to optimize the performance of downstream tasks. However, these methods aim to generate a small set of point clouds, not a subset of point clouds selected from the original point cloud. KCNet [11] and FoldingNet [12] downsample the graph using graph-based maximal pooling, which uses a pre-constructed k-NN graph to obtain the maximum features over the neighborhood of each node. However, these methods do not guarantee that the most essential points are selected to fulfill downstream tasks better.

III. METHODOLOGY

This study provides a methodology for constructing Multi-scale attention dynamic graph convolution (MSADGC) specifically designed for point clouds, providing adequate information for subsequent keypoint detection through feature extraction. Furthermore, the network employs a spatial pyramid feature fusion module to fuse the global features of the downsampled point cloud. The fusion of multi-scale features using spatial pyramid feature fusion (SPFF) can significantly improve the network's capacity to acquire and integrate local and global geometric feature information to optimize downstream tasks. The following sections will further detail the network structure and various modules.

A. Network Structure

The overall network structure of the point cloud deep learning network is shown in Fig. 1. The spatial transformation matrix obtained by training the Spatial T-Net network is first aligned with the input point cloud coordinates and then input into MSADGC to extract the features. The feature information in each layer is extracted, followed by using the self-attention (SA) module to aggregate the features within the local neighborhood. Subsequently, the feature information is aggregated using the spatial pyramid feature fusion, which enables the

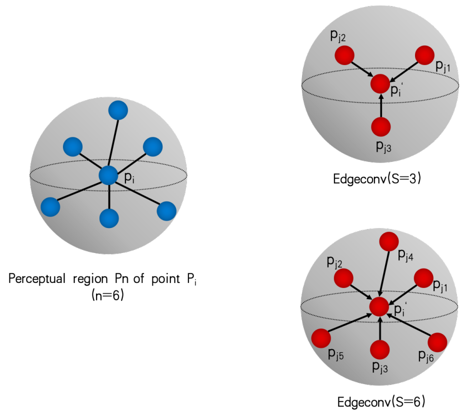

Fig. 2. Left: Pn denotes the n nearest neighbors of the ith point Pi. Right: Visualize the Multi-scale EdgeConv operation.

inclusion of both multi-scale local area features and global features. The classification and segmentation networks then utilize these fusion features for further processing.

B. Multiscale Self-Attention Graph Convolution

While the conventional model directly combines the characteristics of neighboring nodes, the EdgeConv operation employs a nonlinear transformation to extract the embedding information of the edges connecting two interconnected nodes. Then, it fuses edge information with the central node to aggregate the node representation. $G = (V, \mathcal{E})$ represents local point cloud structure, where $V = \{1, ..., n\}$ and $\mathcal{E} \subseteq V \times V$ denote the vertices and edges, respectively. The correlation measure is formally defined as:

$$\boldsymbol{e}_{i,j} = \mathrm{MLP}\left(\boldsymbol{p}_i, \boldsymbol{p}_j - \boldsymbol{p}_i\right) \tag{1}$$

$$\mathbf{p}_i' = \max_{j:(i,j)\in\mathcal{E}} \mathrm{MLP}\left(\boldsymbol{p}_i, \boldsymbol{p}_j - \boldsymbol{p}_i\right) \tag{2}$$

where: $\boldsymbol{p}_j - \boldsymbol{p}_i$ denotes the relative space position of neighboring node; MLP is the multilayer module with nonlinear activation; $\boldsymbol{e}_{i,j}$ and $\mathbf{p}_i'$ present the edge feature and updated node feature respectively.

The MSADGC network consists of multi-scale edge convolutional modules of Fig. 2 and self-attention modules of Fig. 3. The multi-scale edge convolutional modules enable MSADGC to capture the features at different scales of granularity by considering the local structure of each node. Relying on the multi-scale information, MSADGC is adaptable to various graph structures and can perceive different granularity information. The self-attention module allows MSADGC to weightly aggregate the neighboring node messages based on their relevance to the central node, enhancing its ability to handle diverse graph data. The detailed structure of MSADGC is shown in the dashed box of Fig. 1, in which the k-NN graph represents the range of the graph with k nearest neighbor points, two different k-NN Graph algorithms with K taken as 16,32 and edge convolution operation are used to extract different scales of neighborhood feature information respectively, and the two branches perform nonlinear transformations of different dimensions on the local features in parallel to make them more adaptive and expressive.

The designed self-attention module extracts the point cloud context features and combines them with the edge convolution module, which effectively remedies the problem of ignoring the domain-to-domain features in the DGCNN model. The MSADGC network generates refined attentional features based on the global context features by the self-attention module and the local features acquired by the edge convolution module. The MSADGC effectively addresses the issue of inadequate capture of local structural information among points and feature information across neighborhoods and improves the accuracy and robustness of the mode

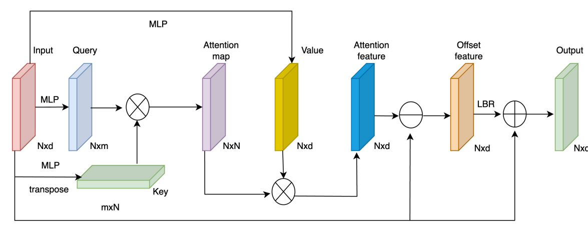

Fig. 3. Details of the self-attention module. LBR combines Linear, BatchNorm and ReLU layers.

C. Spatial Pyramidal Feature Fusion for Point Cloud Downsampling

Channel sampling layer The objective of CSL is to obtain a subset of input points, referred to as Significant Points (SP). During the downsampling process, the critical points of a point cloud contain the most essential information. These points contribute significantly to subsequent downstream tasks.

The input point cloud undergoes feature dimensionality expansion through multiple perceptron layers to obtain an m-dimensional feature vector F_{mid}. Afterward, we use normalization to pull each feature vector onto the same distribution for the features. The maximum value on each point channel dimension can be used as a feature score, representing the main contribution value of the current point. In this case, the network employs a symmetric function $\max_\delta(*)$ for the dimensionality reduction operation. It can be understood that each value in the feature channel represents the probability of the corresponding point being selected as the central point in the next layer. A higher probability indicates a greater likelihood of the point being preserved. Assuming the network needs to sample k important points from this point cloud containing n points, the next step is to simply select the k points with the highest probability values from the feature vector F_{mid}. In the algorithm flow, the operation of finding the indices of the k largest feature values from the n-dimensional vector is denoted as $Top_k(*)$.

The method of sampling the points corresponding to the Top-K maximum value of a channel is called Top K index sampling. The calculation procedure is shown below:

Algorithm 1 Channel Top-K Index Sampling

Input: point set P, and its corresponding feature F;
Output: the point set P' sampling and aggregation, and the corresponding feature F'

function SAMPLING(P, F)
  $\mathbf{F}_{\text{mid}} \leftarrow MLP_{\delta}(\mathbf{F})$
  $\mathbf{F}_{\text{mid}} \leftarrow Softmax\left(\mathbf{F}_{\text{mid}}\right)$
  $\mathbf{F}_{\text{mid}} \leftarrow \max_{\delta}\left(\mathbf{F}_{\text{mid}}\right)$
  $\mathbf{I} \leftarrow Top_k\left(\mathbf{F}_{\text{mid}}\right)$
  $P', F' \leftarrow [P, F, I]$
  return P', F'
end function

Curvature sampling layer An unstructured point cloud's local surface characteristics calculation is a challenging task. Pauly et al. [13] proposed a method that the surface normals and curvature may be intuitively estimated using covariance analysis. To be more specific, considering a point p $\in \mathcal{R}^{NX3}$ along with its set of k-nearest neighbors $\mathbf{N}_p$, the covariance matrix C $\in \mathbb{R}^{3X3}$.

$$C=\left[\begin{array}{c}\mathbf{p}_1-\mathbf{p}\\ \mathbf{p}_2-\mathbf{p}\\ \vdots\\ \mathbf{p}_k-\mathbf{p}\end{array}\right]^T\cdot\left[\begin{array}{c}\mathbf{p}_1-\mathbf{p}\\ \mathbf{p}_2-\mathbf{p}\\ \vdots\\ \mathbf{p}_k-\mathbf{p}\end{array}\right] \tag{3}$$

where $\mathbf{p}_{i=1,k} \in \mathcal{N}_p$.

By performing the eigen-decomposition of the covariance matrix C, the eigenvectors associated with the primary eigenvalues may be obtained. These eigenvectors establish an orthogonal frame located at point $\mathbf{p}$. The eigenvalues, denoted as λ_i, quantify the degree of variability along the axis determined by their respective eigenvector. From an intuitive perspective, the eigenvectors associated with the largest eigenvalues form a basis for the tangent plane at point p, while the eigenvector associated with the smallest eigenvalue can serve as an approximation of the surface normal, denoted as n. Therefore, as the smallest eigenvalue quantifies the deviation of point p from the surface, it can be utilized as an estimation of point curvature. The surface variation $\kappa(\mathbf{p})$ at point p in a neighborhood of size k is defined as follow:

$$\kappa(\mathbf{p})=\frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}, \quad \lambda_0<\lambda_1<\lambda_2 \tag{4}$$

When downsampling in the original point cloud and wanting to retain points that represent the overall structure, by leaving edge points is a good option to retain valid geometric information. The raw point cloud p $\in \mathcal{R}^{NX3}$ and number of nearest neighbor points K fed into the curvature sampling layer.

By doing calculations to determine the curvature value of each individual point and thereafter arranging the points in ascending order based on their curvature values, a subset of K points with higher curvature values may be picked. These points are seen to be more indicative of the edge points within the broader structure. Extracting features from downsampled points and fusing the global features of these points can bring better results to downstream tasks.

TABLE I
CLASSIFICATION RESULTS ON MODELNET40.

Method	Overall Accuracy
PointNet [1]	89.2%
PointNet++ [2]	91.9%
SpiderCNN [16]	92.4%
DGCNN [3]	92.9%
PointCNN [5]	92.2%
PointConv [23]	92.5%
KPConv [6]	92.9%
PointASNL [18]	93.2%
PT [19]	92.8%
PCT [20]	93.2%
PRA-Net [21]	93.7%
PAConv [22]	93.6%
MGE-Net	**93.9**%

IV. EXPERIMENTAL RESULTS

A. Implement Details

We evaluate the model with the Pytorch framework on a single RTX 3090 GPU with 16G memory. To train the model, we use AdamW optimizer with an initial learning rate 1×10^{-3} and decay it to 1 $\times10^{-8}$ with a cosine annealing schedule. The weight decay hyperparameter is set to 1×10^{-4}. Dropout with a probability of 0.5 is used in the last two fully connected layers. We train the classification network with a batch size of 8 for 200 epochs and the segmentation network on a scale of 16 for 200 epochs.

B. Classification

Dataset. Experiments for the point cloud classification task were performed using the ModelNet40 dataset [14]. The dataset consists of 12,311 CAD models categorized into 40 object classes. It is split into a training set comprising 9,843 models and a test set with 2,468 models. The experiments in this paper followed the same division ratio, and the classification results were reported based on the test set.

Quantitative and Qualitative Results. Tab. I summarizes the quantitative comparison with SOTA methods, where our proposed approach is the best method. The qualitative results are shown in Fig. 4. From the first line of images, we can observe that the key points were sampled, while the second line shows a better edge sampling effect. Sampling more representative points can greatly improve downstream tasks

C. Part Segmentation

Dataset. For the task of point cloud segmentation, the ShapeNetParts dataset [15] is utilized for conducting experiments. This dataset comprises 16 object categories and 50 part segmentation labels, encompassing 16,881 3D models. The evaluation metric employed is instance mIoU (mean Intersection over Union), which quantifies the quality of segmentation results.

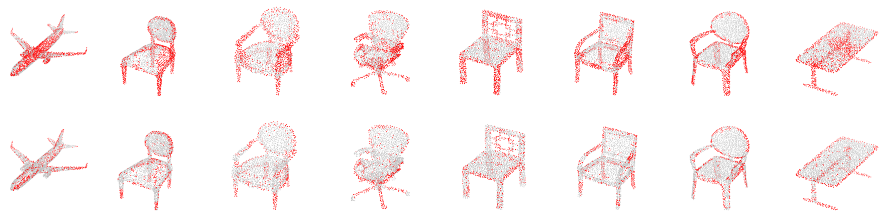

Fig. 4. Visualized sampling results of Channel sampling and Curvature sampling on different shapes.

TABLE II
SEGMENTATION RESULTS ON SHAPENET PART.

Method	Ins. mIoU
PointNet [1]	83.7%
PointNet++ [2]	85.1%
SpiderCNN [16]	85.3%
DGCNN [3]	85.2%
SPLATNet [17]	85.4%
PointConv [23]	85.7%
PointCNN [5]	86.1%
KPConv [6]	86.2%
PT [19]	85.9%
PCT [20]	86.4%
PRA-Net [21]	86.3%
PAConv [22]	86.1%
MGE-Net	85.9%

TABLE III
SEGMENTATION RESULTS OF THE FULL POINT CLOUDS AND INTERMEDIATE DOWNSAMPLED POINT CLOUDS OF DIFFERENT SIZES.

Method \ Points	2048	1024	512	256	128
MGE-Net	85.86%	87.29%	88.23%	89.15%	90.41%

Quantitative and Qualitative Results. Tab. II lists the segmentation quantitative results, and our method has achieved good performance, but cannot be compared to the best method. However, when we calculated the same metrics for the intermediate downsampling point cloud in Tab. III, we were surprised to find that their performance was very good, even far superior to the SOTA method. This indicates that downsampling more key points contributes more to performance. Fig. 5 also shows the qualitative segmentation results of the downsampling process, indicating that edge key points with important information are retained during the downsampling process.

D. Ablation study

In this subsection, multiple ablation studies are conducted regarding the design choices of neural network architectures.

Feature Learning Layer. The feature learning layer we used in the above experiments is the MSADGC Embedding layer. We additionally report the results of using EdgeConv as the feature learning layer in Tab. IV. From it, we can observe that MSADGC achieves the best performance. Meanwhile, the results of using EdgeConv are improved when using our proposed sampling methods.

TABLE IV
ABLATION STUDY OF USING DIFFERENT FEATURE LEARNING LAYERS

Method	Feature Learning Layer	OA(%)	mIoU(%)
DGCNN	EdgeConv	92.90	85.20
MGE-Net	EdgeConv	93.51	85.54
	MSADGC	93.88	85.86

TABLE V
ABLATION STUDY OF USING A DIFFERENT NUMBER OF EMBEDDING DIMENSIONS

Method	Embedding Dimension	OA(%)	mIoU(%)
MGE-Net	64	93.39	85.80
	128	93.88	85.86
	192	93.60	86.24

Embedding Dimension. In our experiments, we used 128 as the default embedding dimension. We also report results using embedding dimensions of 64 and 192 in Tab. V.

Evaluation on k in Curvature Sampling. The choice of the number of neighbors, denoted as k, is a crucial parameter in curvature sampling, as it directly impacts the extent of the local perceptual region. Additionally, we present findings indicating that employing various values of k yields different outcomes in Tab. VI.

CONCLUSION

This paper proposes an adaptive key point cloud sampling method that combines neural network-based learning and direct point selection based on local geometry estimation. At the same time, we have designed a feature extraction module based on multi-scale attention graph convolution to provide richer feature information for subsequent key point detection. Our method downsampling the input point cloud to any desired

TABLE VI
ABLATION STUDY OF USING A DIFFERENT NUMBER OF NEIGHBORS FOR POINT SAMPLING IN CURVATURE SAMPLING LAYER.

k	8	16	32	64	128	256
OA(%)	93.59	93.88	93.76	93.56	93.35	93.55
mIoU(%)	85.85	85.86	85.89	85.91	85.89	85.85

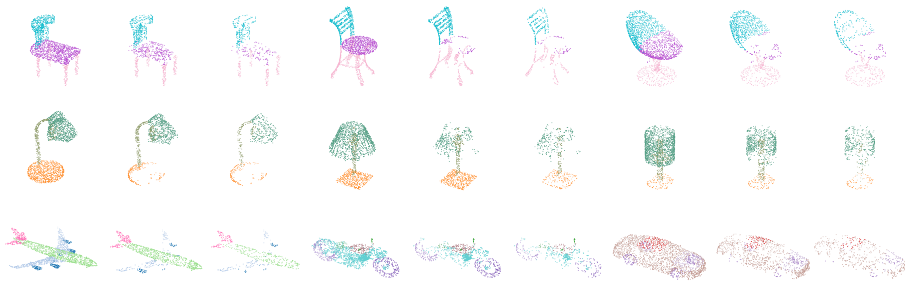

Fig. 5. Visualized segmentation results as shape point clouds are downsampled.

size. This feature makes our method highly efficient and practical in processing large-scale point cloud data. Good qualitative and quantitative results have been achieved on the common point cloud benchmark, demonstrating the effectiveness of the proposed sampling method.

ACKNOWLEDGEMENT

Our work is supported by the Research Development Funding of Xi'an Jiaotong-Liverpool University (No.RDF-21-02-080).

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652-660.

[2] C. R. Qi, L. Yi, H. Su, L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in neural information processing systems, 30.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, 'Dynamic Graph CNN for Learning on Point Clouds', ACM Trans. Graph., vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[4] C. Chen, L. Z. Fragonara, and A. Tsourdos. Gapnet: Graph attention based point neural network for exploiting local feature of point cloud. Neurocomputing, 438:122–132, 2021. 2

[5] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, 'PointCNN: Convolution On $\mathcal{X}$-Transformed Points'. arXiv, Nov. 05, 2018. Accessed: Oct. 25, 2023. [Online]. Available: http://arxiv.org/abs/1801.07791.

[6] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, 'KPConv: Flexible and Deformable Convolution for Point Clouds', in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South): IEEE, Oct. 2019, pp. 6410–6419. doi: 10.1109/ICCV.2019.00651.

[7] Z. Lin, S. Huang and Y. Wang, "Convolution in the Cloud: Learning Deformable Kernels in 3D Graph Convolution Networks for Point Cloud Analysis," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1797-1806, doi: 10.1109/CVPR42600.2020.00187.

[8] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. In Eurographics, 2003. 1, 2

[9] O. Dovrat, I. Lang and S. Avidan, "Learning to Sample," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019 pp. 2755-2764. doi: 10.1109/CVPR.2019.00287

[10] I. Lang, A. Manor, and S. Avidan, 'SampleNet: Differentiable Point Cloud Sampling', in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2020, pp. 7575–7585. doi: 10.1109/CVPR42600.2020.00760.

[11] 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, volume 4, 2018.

[12] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud autoencoder via deep grid deformation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 3, 2018.

[13] M. Pauly, M. Gross and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," IEEE Visualization, 2002. VIS 2002., Boston, MA, USA, 2002, pp. 163-170, doi: 10.1109/VISUAL.2002.1183771.

[14] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d ShapeNets: A deep representation for volumetric shapes," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015, pp. 1912–1920.

[15] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A Scalable Active Framework for Region Annotation in 3d Shape Collections," ACM Trans. Graph., vol. 35, no. 6, pp. 210:1–210:12, Nov. 2016.

[16] Y. F. Xu, T. Q. Fan, M. Y. Xu, L. Zeng, and Y. Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. ArXiv, abs/1803.11527, 2018. 5, 6

[17] H. Su, V. Jampani, D. Q. Sun, S. Maji, E. Kalogerakis, M. H. Yang, and J. Kautz. Splatnet: Sparse lattice networks for point cloud processing. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2530–2539, 2018. 6

[18] X. Yan, C. D. Zheng, Z. G. Li, S. Wang, and S. G. Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5588–5597, 2020. 1, 5

[19] N. Engel, V. Belagiannis and K. Dietmayer, "Point Transformer," in IEEE Access, vol. 9, pp. 134826-134840, 2021, doi: 10.1109/ACCESS.2021.3116304.

[20] M. H. Guo, J. X. Cai, Z. N. Liu, T. J. Mu, R. R. Martin, and S. M. Hu. Pct: Point cloud transformer. Comput. Vis. Media, 7:187–199, 2021. 2, 5, 6

[21] S. L. Cheng, X. W. Chen, X. W. He, Z. Liu, and X. Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. IEEE Transactions on Image Processing, 30:44364448, 2021. 5, 6

[22] M. T. Xu, R. Y. Ding, H. S. Zhao, and X. J. Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3172–3181, 2021. 2, 5, 6

[23] W. X. Wu, Z. G. Qi, and F. X. Li. Pointconv: Deep convolutional networks on 3d point clouds. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9613–9622, 2019. 1, 2, 5, 6

[24] Y. Li and G. Baciu. Hsgan: Hierarchical graph learning for point cloud generation. IEEE Transactions on Image Processing, 30:4540–4554, 2021.2

[25] Y. Li and G. Baciu. Sg-gan: Adversarial self-attention gcn for point cloud topological parts generation. IEEE Transactions on Visualization and Computer Graphics, 28(10):3499–3512, 2021.2

[26] M. Zhang., Y. Li, R. Chen, Y. Pan, J. Wang, Y. Wang, and R. Xiang (2024). WalkFormer: Point Cloud Completion via Guided Walks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3293-3302).