Combine Local and Global Feature Extraction for Point Cloud Classification

Xiaolong Lu, Baodi Liu, Weifeng Liu, Kai Zhang, Ye Li, and Peng Liu

Abstract The point cloud is one of the common formats of 3D data, and it can represent the shape of objects more intuitively. However, due to the point clouds' irregularity and disorder, there are still many problems during processing. The previous approach was to convert point clouds to other formats for processing until PointNet came along, which pushed point cloud data directly into network processing for the first time, achieving a breakthrough. The conventional approaches to dealing with point clouds rarely simultaneously consider the global and the local features of point clouds. With the appearance of attention mechanism and graph structure, the application on point cloud also has a certain effect. In particular, the graph structure is more suitable for the processing of the point cloud due to its characteristics. In this paper, the attention mechanism is the basis to enhance the representation of nodes, and then the dynamic graph and point network are fused to extract local and global features, respectively. Finally, we conducted experimental verification on the benchmark datasets, such as ModelNet40 and ScanObjectNN, and achieve superior performance to several state-of-the-art approaches.

Keywords Point cloud classification · Attention mechanism · Dynamic graph · Local and global feature

X. Lu
College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China

B. Liu (✉) · W. Liu
College of Control Science and Engineering, China University of Petroleum (East China), Qingdao, China
e-mail: liubaodi@upc.edu.cn

P. Liu
Shandong Kexun Information Technology Co., Ltd., Qingdao, China

Y. Li
Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

K. Zhang
School of Petroleum Engineering, China University of Petroleum (East China), Qingdao, China

J. Yao et al. (eds.), *The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021)*, Lecture Notes in Electrical Engineering 813,
https://doi.org/10.1007/978-981-16-6963-7_45

1 Introduction

With the increasing application of 3D data, a series of sensors, such as lidar sensors, emerge at the historical moment, which acquires 3D data more efficiently and effectively. As the most representative point cloud data in 3D data, because it can best reflect the original sensor data, it has become more and more important, and its use has also increased. It has become particularly popular in areas such as autonomous driving, robot recognition, and unmanned aerial vehicles. However, due to the irregularity, disorder, and sparsity of point cloud data, there are certain challenges in actual processing.

The traditional work is to transform point cloud data into other data formats: Based on multi-view and volumetric grid.

For Multi-view, the MVCNN [2] is a pioneering work based on multi-view. Its innovation is to use 2D rendering images obtained from 3D data of objects from different "perspectives" as original training data and then apply a 2D convolution operation training model to achieve a better classification effect. MHBN [3] is used to coordinate bilinear pools to aggregate local convolution features to obtain an effective representation of 3D objects. For the volumetric grid, VoxNet [1] uses 3D CNN to process the voxel of the occupied grid, which can2 quickly and accurately classify 3D data. In order to make a 3D convolutional neural network effective in existing models, OctNet [4] divides sparse data into a series of hybrid octrees by layers, which enables the network to achieve both deep level and high resolution. PointGrid [5], which is a three-dimensional convolutional network, is the integration of points and grids and is a hybrid model that can better represent the details of local geometry. But in the process of conversion, important data information is easily lost, and unnecessary data redundancy is introduced, which increases the computational burden.

With the development of scientific research technology, PointNet [6] emerged, making direct use of point cloud data. This network uses multi-layer perceptron and symmetric functions to ensure the disorder and permutation invariance of the point cloud. Since the network is trained to point by point, it does not consider the local feature extraction. Later PointNet++ [7] is optimized based on PointNet, and the most distant point sampling method is adopted to realize hierarchical feature extraction so as to obtain fine-grained features from the neighborhood of each point. But PointNet++ only uses local ball query to do max pooling, and there is no deeper exploration of the local feature information of the point cloud.

In order to solve the problem that the information between points cannot be considered, the graph structure is introduced into point cloud processing. DGCNN [8] first uses k-NN to construct the graph structure between points, analyzes the geometric relations between points more accurately, and then uses the edge convolution module to fully extract the local information. Further, LDGCNN [17] removes the transformation network in DGCNN and is inspired by Densenet to connect different levels of features and improve the network's performance. Zhang et al. [9] proposed a new network based on graph convolution, which mainly used the graph convolution form

of Chebyshev polynomial to process the point cloud features and finally realized the classification through the full connection layer. Due to the uniqueness of the point cloud, the effectiveness of attention mechanisms in various fields, and the influence of graph attention mechanism, GAPNet [10] introduced graph attention mechanism into the point cloud processing task. However, in the process of utilization, only point-based network feature extraction was used in the later processing, which could not extract more fine-grained features.

In this paper, to overcome the inadequacies of point cloud feature extraction, we extract the attention feature based on self-attention and neighboring-attention and then extract the fine-grained feature of the point cloud from local and global aspects so that the context information of point cloud and the global information of point cloud can be fully considered. The 3D coordinates of the point cloud are taken as input and then through the transformation network with attention. On the one hand, the dynamic graph network is used to further process the point cloud, which enables the network to extract more detailed local neighborhood information. On the other hand, let the point cloud go through a multi-head parallel attention mechanism and then through a stacked multi-layer perceptron (MLP) with Shared parameters to extract global features. Finally, a graphic-based pooling layer is connected to further enhance the robustness of the network.

The main contributions of this paper are as follows:

(1) It is easy to ignore the structural information between dropped points when only considering the feature extraction of points. In this paper, global and local features are considered at the same time so that more fine-grained information can be mined.
(2) In this paper, on the basis of including the attention mechanism, we combine the dynamic graph structure with the Shared perception machine module with jump connection to get a better effect.
(3) We tested our network on the classified data sets ModelNet40 and ScanObjectNN and showed the performance of the network in the experimental part.

2 Methodology

In this section, we describe the architecture that our network uses to better handle irregular data, such as point clouds, which are mainly used for point cloud classification processing tasks. We explained the method used in this paper in detail, mainly including the following three parts: attention mechanism module, dynamic graph structure, point network structure, and the network model we designed is shown in Fig. 1.

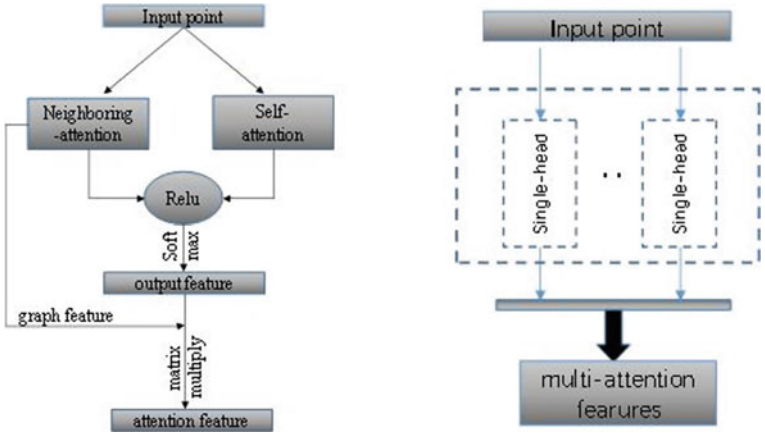

Fig. 1 *GAPLayer.* For the left part, the single-headed GAPLayer first extracts the self-attention and neighboring-attention features of the point cloud, respectively, and then performs a normalization operation through a nonlinear activation function LeakyRelu and softmax functions, and finally gets the final attention characteristics by multiplying the graph features. The right part represents a multi-headed attention mechanism, and 4-heads are used in this paper

2.1 Attention Mechanism Module

The attention machine used in this paper considers the local and global structure and uses the self-attention mechanism and the neighboring-attention mechanism, as shown in the left part of Fig. 1. Through the self-attention mechanism, the importance of independent points is better analyzed, and corresponding weight information is given, which is represented by Eq. 1:

$$x_i' = g(x_i, \theta) \tag{1}$$

where, x_i' represents the output features encoded on the point features, g() represents a nonlinear function (we use a multi-layer perceptron with shared parameters), while θ represents a learnable filter parameter. For the neighbor attention module, we first use the k-NN (k-nearest neighbor algorithm) to construct a graph structure G = (V, E) to represent the relationship between points, where V = (1, 2, . . . , N) is the number of points of a point cloud object, E is the edge information connecting adjacent points. We defined the feature information of this part as $y_{ij} = (x_i - x_{ij})$, where i and j respectively represent node and neighbor point indexes, x_{ij} is the neighboring point x_j to point x_i. Similar to Eq. 1, this part can be expressed by the following formula:

$$y_{ij}' = g\left(y_{ij}, \theta\right) \tag{2}$$

Through the above two formulas, we can obtain the self-attention feature x_i' and the neighboring-attention feature y_{ij}', the attentional features can then be obtained

from Formula 3, where LeakyReLU() represents a nonlinear activation function.

$$c_{ij} = LeakyReLU\left(g\left(x_i', \theta\right) + g\left(y_{ij}', \theta\right)\right) \tag{3}$$

To better explore the attention features of the point cloud and more local and global information, we also constructed a multi-head attention structure (Fig. 1 (right)) based on the above attention module and ran the multi-head attention module in parallel.

2.2 Feature Extraction from Point Network

In this section, we first apply the multi-head attention mechanism mentioned in the previous section to obtain features with attention coefficients. Inspired by PointNet [6], the feature with the attention coefficient is input into an MLP with shared parameters to extract the fine-grained feature at the point. In the process of its implementation, ResNet [11] is used for reference, and a skip connection is adopted, which makes the input contain more layers of information. To improve the network's performance, we also introduce an attention-pooling operation to identify the most important part of the multi-head attention feature.

2.3 Feature Extraction from the Graph Structure

We know that point-based feature extraction cannot capture the context information of the point cloud well, nor can it achieve a better classification effect. Inspired by DGCNN [8], we fuse the dynamic graph structure better to connect the context information of the point cloud and facilitate the extraction of fine-grained features.

In this part, we first construct a directed graph G = (V, E) to represent the local structure between point clouds, where V = (1, 2, …, N) and E ⊆ V × V indicate vertex and edge information about a point cloud, respectively. At the same time, the edge feature is defined as $e_{ij} = g_\Theta(x_i, x_j)$, where g : $R^C \rightarrow R^{C'}$ represents a nonlinear function with a set of trainable parameters. Finally, a convolution operation is defined on the graph, and an asymmetric channel-based aggregation function is used (for example, Σ or pooling). Simply put, input a 3D point cloud with n points, and this operation generates a nC'-dimensional point cloud with n points. Therefore, the graph convolution output of the i-th node can be expressed by the following formula:

$$x_i' = \underset{j:(i,j)\in\varepsilon}{\Delta} g_\Theta\left(x_i, x_j\right) \tag{4}$$

The choice of the nonlinear function g and the aggregate function Δ is also crucial. In the previous approach, some researchers tried to encode only the global information and ignore the local structure information, such as the method in PointNet. Later, some people said that the information was divided into small pieces, which fully considered the local information but lost the global information. In this work, we calculate the global structure information and local structure information through the coordinate information of and $x_j - x_i$ respectively, and the output of one layer can be expressed by the following formula:

$$g_\Theta(x_i, x_j) = \overline{g}_\Theta(x_i, x_j - x_i) \tag{5}$$

2.4 *Our Network*

Inspired by GAPNet [10] and DGCNN [8], we proposed the model of this paper, as shown in Fig. 2. In this model, we first use a 4-head attention module to deal with the point cloud so that the output of the transformation network has certain attention

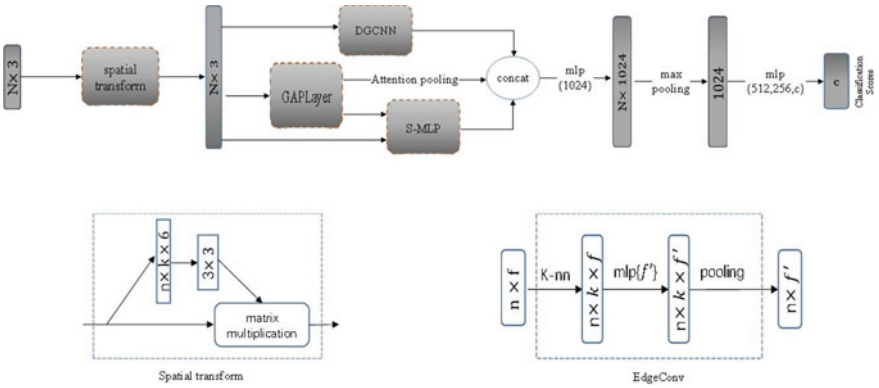

Fig. 2 *The network model:* The proposed network model is mainly used for classification tasks. The classification model takes N as input and only considers the 3D coordinate (x, y, z) of the point cloud. The upper part uses a dynamic graph module to extract the local information of the point cloud. And the lower part first uses a GAPLayer to obtain an attention feature and graph feature. Here, multi-graph features are followed by an attention pooling operation, and multi-attention features are followed by a perceptron with shared parameters to extract point-based features. Finally, global features, local features, and attention pooling features are combined to obtain the classification score of category c. *Spatial transformation network:* The use of this network guarantees the permutation invariance of the point cloud. The model first uses a single-head attention module and finally generates a 3 × 3 transformation matrix. *Edge convolution:* This module is mainly a structure used in dynamic graphs (DGCNN). First, a graph structure is constructed using k-NN. Then edge feature information is calculated through the perceptron module with shared parameters and pooling operations

features and keeps some transformation invariance to the point cloud. Next, a multi-head attention module is used to extract feature attention information and apply it to a multi-layer perceptron with a jump connection structure to extract more fine-grained features. At the same time, we also introduce a dynamic graph structure and run it in parallel with the above parts to extract local information between point clouds, thus ensuring that our model carefully considers both local and global information. Finally, an attentional pooling operation is introduced to make the entire network model more robust and improve performance.

3 Experiments

In this part, we first introduce two point cloud datasets commonly used for classification and verify our model on these two datasets. Then our method is compared with existing methods in the CAD model and real-world point cloud dataset. Finally, we introduce the ablation experiment to prove the strong performance of our network.

Datasets. We experimented with two datasets and demonstrated the performance of our network: a synthetic CAD model classification dataset ModelNet40 [19] and a real-world point cloud dataset ScanObjectNN [20].

ModelNet40. As a classical point cloud classification data set, its application in recent years has also received great attention. The ModelNet40 data set contains 12,311 synthetic grid CAD models, which are composed of 40 categories. Among them, the training data contains 9843 models, and the remaining models are used for experimental testing. In addition, 1024 points were uniformly sampled on the surface of each model, and the model was further normalized through the unit sphere. For simplicity, the 3D coordinates (x, y, z) of the point cloud were only used in the experiment of this paper.

ScanObjectNN. Experiments on synthetic datasets have achieved relatively high performance; however, experiments on real-world point cloud datasets remain a challenge. In this paper, we have done experiments on the ScanObjectNN data set and made a comparison. The data set has about 15,000 objects, divided into 15 categories of information. Since the data set reflects the point cloud data in the real world, there are some missing parts and the influence of the background, which makes the data set according to the challenge.

Training. We use adam optimization model with a momentum of 0.9, and the learning rate is set to 0.001, and the batch size is 32 and trained 250 epochs. The decay rate of batch normalization was initially 0.7 and gradually increased to 0.99 during training. Our model is trained on Nvidia Tesla V100 GPU and TensorFlow-GPU 1.14.

Experimental results on ModelNet40. Experimental results on the dataset are shown in Table 1. This table shows several recently popular point cloud classification methods. It is obvious that our method has better experimental results than

Table 1 Experimental results on ModelNet40

Method	Avg class acc. (%)	Overall acc. (%)
ECC [12]	83.2	87.4
PointNet [6]	86.0	89.2
PointNet+ + [7]	87.8	90.7
SO-Net [13]	87.3	90.9
KCNet [14]	–	91.0
3DmFV [15]	86.3	91.4
DGCNN [8]	90.2	92.2
GAPNet [10]	89.7	92.4
Ours	**90.3**	**92.7**

the compared methods, which is 0.5% higher than DGCNN and 0.3% higher than GAPNet. The symbol '–' means the results are unavailable.

Experimental results on ScanObjectNN. With the same experimental setup, we also verified our model on ScanObjectNN, the real-world dataset, as shown in Table 2. As can be seen from the table, the overall accuracy of our experiment has reached 80.7%, which is significantly better than other methods. This experiment fully demonstrates that the effectiveness and robustness of our model in the face of a real-world dataset are exactly what we need now. In Fig. 3, we show the accuracy of each category and can clearly see that our proposed method is better than the other methods compared through experiments.

Ablation studies. In order to verify the validity of our model, we also carried out ablation experiments on ModelNet40 and ScanObjectNN, respectively, as shown in Table 3. On ModelNet40, we compare the proposed method with that in garnet. It can be seen from the table that the accuracy of the proposed method is reduced by 0.2% when the original attention pooling layer is replaced by a dynamic graph. When we run dynamic graphs in parallel, our experimental results are improved by 0.3%. Ablation studies on ScanObjectNN are shown in the third column of the table.

Table 2 Experimental results on ScanObjectNN

Method	Avg class acc. (%)	Overall acc. (%)
3DmFV [15]	58.1	63
PointNet [6]	63.4	68.2
SpiderCNN [16]	69.8	73.7
PointNet+ + [7]	75.4	77.9
DGCNN [8]	73.6	78.1
PointCNN [18]	75.1	78.5
Ours	**77.8**	**80.7**

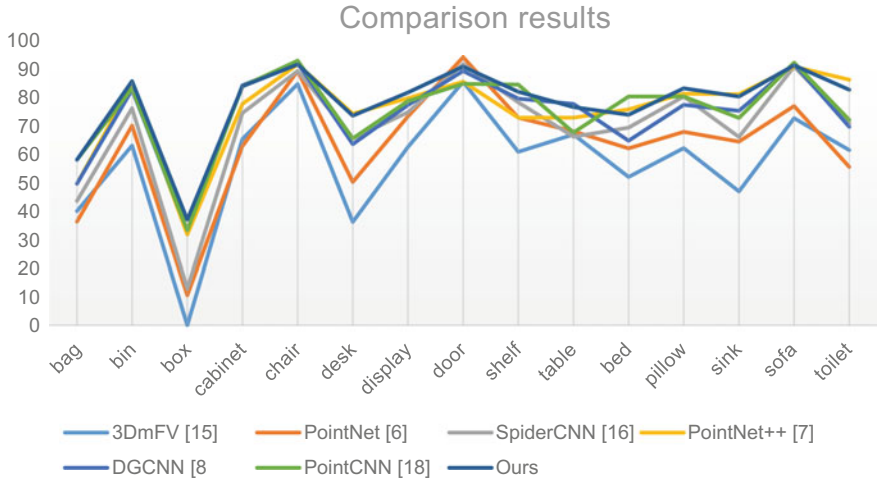

Fig. 3 Comparison of the accuracy of each class on the dataset ScanObjectNN

Table 3 Ablation experiments

Components	ModelNet40	ScanObjectNN
GAPlayer + attention pooling	92.4	76.2
GAPlayer + DGCNN	92.2	80.4
GAPlayer + DGCNN + attention pooling	**92.7**	**80.7**

4 Conclusion

In this paper, we propose a new model for the point cloud classification task. We fully consider the local and global information of the point cloud and extract the information through the dynamic graph structure and the point network structure with attention features, respectively. Moreover, the method in this paper is still based on the graph attention mechanism module, and the extracted features with attention coefficient are further processed. Finally, an attention pooling operation is introduced to ensure the effectiveness and robustness of the network. Our model is validated on ModelNet40 and ScanObjectNN datasets and compared with several advanced methods, and the experimental results prove that our model has better results and more robust performance. In the future, we can consider using our model in large point cloud tasks such as semantic segmentation.

Acknowledgements The paper was supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2019MF073), the Open Research Fund from Shandong Provincial Key Laboratory of Computer Network (No. SDKLCN-2018-01), the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) (Grant No. 20CX05001A),

the Major Scientific and Technological Projects of CNPC (No. ZD2019-183-008), and the Creative Research Team of Young Scholars at Universities in Shandong Province (No.2019KJN019).

References

1. Daniel, M., Sebastian, S.: Voxnet: A 3D convolutional neural network for real-time object recognition. In: International Conference on Intelligent Robots and Systems (IROS), pp. 9–922. IEEE (2015)
2. Hang, S., Subhransu, M., Evangelos, K., Erik, L.: Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 9–945 (2015)
3. Tan, Y., Jingjing, M., Junsong, Y.: Multi-view harmonized bilinear network for 3D object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 186–194 (2018)
4. Gernot, R., Ali, O., Andreas, G.: Octnet: Learning deep 3D representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 35–3577 (2017)
5. Truc, L., Ye, D.: Pointgrid: A deep network for 3D shape understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 92–9204 (2018)
6. Charles, R.Q., Hao, S., Kaichun, M., Leonidas, J.: Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6–652 (2017)
7. Charles, R.Q., Li, Y., Hao, S., Leonidas, J.: Pointnet++: deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems (NeurIPS), pp. 510–5099 (2017)
8. Yue, W., Yongbin, S., Ziwei, L., Sanjay, E.S., Michael, M.B., Justin, M.S.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. (tog) **38**(5), 1–12, 201 (2019)
9. Yingxue, Z., Michael, R.: A graph-CNN for3d point cloud classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6–6279. IEEE (2018)
10. Can, C., Luca Zanotti, F., Antonios, T.: Gapnet: graph attention based point neural network for exploiting local feature of point cloud. arXivpreprint arXiv: 1905.08705 (2019)
11. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7–770 (2016)
12. Martin, S., Nikos, K.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 37–3693 (2017)
13. Jiaxin, L., Ben, M.C., Gim Hee, L.: So-net: self-organizing network for point cloud analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 94–9397 (2018)
14. Yiru, S., Chen, F., Yaoqing, Y., Dong, T.: Mining point cloud local structures by kernel correlation and graph pooling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 45–4548 (2018)
15. Yizhak, B., Michael, L., Anath, F.: 3DMFV: threedimensional point cloud classification in real-time using convolutional neural networks. IEEE Robot. Autom. Lett. **3**(4), 3145–3152 (2018)
16. Yifan, X., Tianqi, F., Mingye, X., Long, Z., Yu, Q.: SpiderCNN: deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–87 (2018)

17. Kuangen, Z., et al.: Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features. arXiv preprint arXiv: 1904.10014 (2019)
18. Yangyan, L., Rui, B., Mingchao, S., Wei, W., Xinhan, D., Baoquan, C.: PointCNN: convolution on x-transformed points. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 8–820 (2018)
19. Zhirong, W., Shuran, S., Aditya, K., Fisher, Y., Linguang, Z., Xiaoou, T., Jianxiong, X.: 3D shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 192–1912 (2015)
20. Mikaela Angelina, U., Quang-Hieu, P., Binh-Son, H., Duc Thanh, N., Sai-Kit, Y.: Revisiting point cloud classification: a new benchmark dataset and classification model on real-world data. arXiv, pp. 20–908 (2019)