3D Object Recognition Based on Point Cloud Geometry Construction and Embeddable Attention

No Author Given

No Institute Given

Abstract. A point cloud is a collection of disordered and discrete points with irregularity, and it lacks of topological structure. The number of discrete points in the point cloud is huge, and how to capture the key features from the large amount of points is crucial to improve the accuracy of model recognition. In this paper, based on point cloud geometry construction and embeddable attention, a 3D object recognition algorithm is proposed. By constructing triangular geometries between points, topological structure information to the point cloud is stored for points' geometric construction module. The embeddable attention module uses an improved attention mechanism with feature bias and nonlinear mapping to enable focused attention to capture key features. In addition, a combination of max and average pooling to aggregate global feature has been applied to avoid situations when using only one method would ignore other key information. In comparison with other state-of-the-art methods using ModelNet40 and ScanObjectNN, the proposed method shows significant improvements in identifying both mAcc and OA. The experiments also demonstrate the effectiveness of the modules in this algorithm.

Keywords: 3D object recognition · Point cloud · Convolutional Neural Network · Geometric construction · Embeddable attention.

1 Introduction

In recent year, 3D object recognition has become a research hotspot in the field of computer vision with great research prospects. Such as autonomous driving [1], intelligent robotics [2], virtual reality [3], etc.

The research of classification in pattern recognition based on point cloud has received wide attention from researchers around the world, and the accuracy of classification in pattern recognition has been significantly improved over the past years. However, due to the fact that each point cloud model contains a large number of discrete points, it may cause information overload. In addition, point clouds are aggregated from disordered discrete points without topological structure, and it lacks geometric information. According to the analysis of the above deficiencies, this paper proposes a 3D object recognition algorithm based on point cloud geometry construction and embeddable attention. The algorithm mainly includes three steps, i.e., point cloud data pre-processing, feature extraction and classification in pattern recognition. In the stage of point cloud pre-processing, the point cloud is randomly sampled from a uniform distribution to reduce the number of discrete points in each point cloud to avoid information redundancy and to simplify the computational complexity. Feature extraction is the key step, which gradually expands the perceptual field through local feature extraction and finally aggregates to obtain a global shape descriptor with rich semantic information. Finally, the class of each point cloud model is obtained by the classifier.

In summary, the main contributions of this paper are as follows:

- Point geometry construction module (PGCM) is proposed to construct triangular geometric structures for sampled points and their two nearest neighboring points. The topological structure information is attached to the point cloud to make up for the shortage of geometric information, so that the extracted point cloud shape descriptors are closer to the real shape of 3D objects.
- During the deep feature extraction of point clouds, an embeddable attention module (EAM) is introduced to achieve focused attention on key information. Meanwhile, the module is embeddable and can be ported to other network structures. By applying a combination of max pooling and average pooling, global features are aggregated for avoiding situations where critical information is overlooked.
- According to the experimental results, the proposed method shows significant improvements over other advanced algorithms from point cloud benchmark datasets ModelNet40 and ScanObjectNN.

The rest of this paper is structured as follows. A brief review of the related work is presented in Section 2. Then, the various parts of the proposed algorithm are presented in Section 3. Section 4 presents the specific configuration of the experiments and the comparison of the experimental results with other algorithms. Finally, the conclusions of this paper are presented in Section 5.

2 Related Work

Along with the iterative update of GPU computing power and the emergence of large 3D model data in the computer field, point cloud classification methods based on deep learning have gradually taken a dominant position, and in this section we mainly introduce several point cloud classification methods related to our algorithm.

2017 Qi et al [4] pioneered the PointNet which is directly applied to point cloud learning, which learns features of individual points by MLP and uses symmetric function max pooling to solve the disorder of point clouds; A three-dimensional spatial transformer network is used to solve the problem of rotation invariance of point clouds; Geometric and feature transformations are performed on the input point clouds, and max pooling aggregated point features are used to solve the problem of point cloud substitution invariance. Although PointNet provides a new idea for learning point clouds, it only captures the information of individual points and global points when extracting features, and does not fully consider the interaction of neighboring points and does not extract the local shape information. Without knowing the local shapes, it is difficult for point cloud learning. To learn from local structures, we define a point cloud geometric construction module with discrete and explicit locality, with additional triangular geometry shape information to complement the Cartesian coordinate information of the points.

Convolutional Neural Networks (CNN) have also achieved good results in the classification and recognition task of point clouds. Although traditional CNNs can exploit spatial local correlation, applying them directly to irregular point clouds will not only lose point cloud shape information, but also suffer from point cloud disorder. Li et al [7] proposed the X-transformation transform convolution operator in PointCNN, which solves the problem of disorder of point clouds to some extent. The KPConv proposed by Thomas H. et al [9] provides variable convolution operators that use a set of kernel points to define the region applied to each kernel weight. In the proposed method, we use symmetric function max pooling to solve the disorder of point clouds, and apply a 1×1 convolution operation along with batch normalization and activation functions to implement MLP. Other researchers have applied attention mechanisms to point cloud classification. Attentional mechanisms enable the system to focus on primary information and ignore secondary information. Guo et al [11] proposed a point cloud transformer network (PCT) for point cloud learning by borrowing the transformer structure in the field of natural language processing, and used stacked offset attention modules in the encoder part to improve the accuracy of classification and recognition of point clouds. In this paper, we also propose an embeddable attention module that uses feature bias and nonlinear mapping to improve the self-attention mechanism. The key information is focused and the network model is optimized.

3 Deep Hierarchical Network for Point Clouds

3.1 Point Geometric Construction Module

In contrast to existing research programs, we intend to provide some clues about the low-level geometry for the network, rather than repeating similar information for each layer. Point cloud data, although easy to collect, lacks geometric information compared to well-constructed mesh or voxel data. To remedy this drawback and better capture the geometric feature information of point clouds for representing 3D objects adequately, we explicitly enrich the geometric information of points in the low-level space, and apply the MLP-based hierarchy in the high-level feature space to implicitly learn the local geometric context of points and the global feature information between points. For a particular 3D object, we only provide the coordinate information of the 3D point cloud model as a priori knowledge of the network. However, this is not sufficient to describe

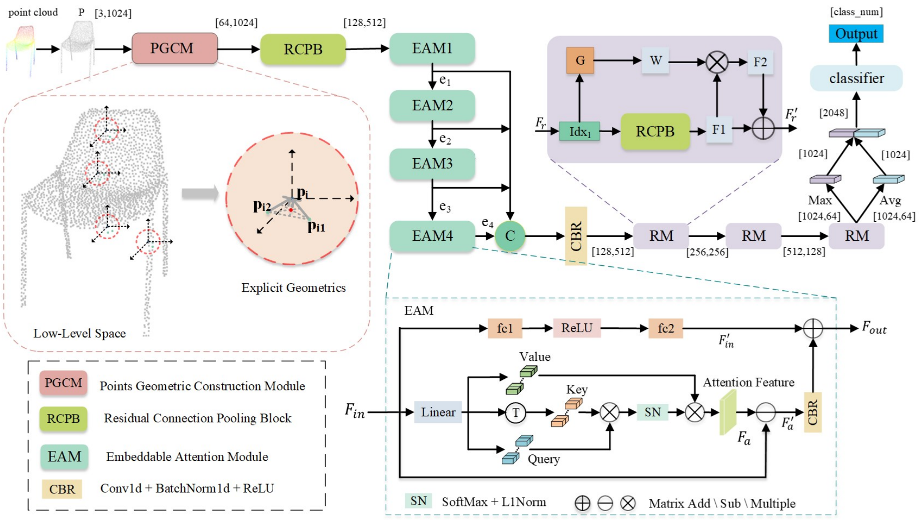

Fig. 1. Network structure of our algorithm. 1024 points are randomly and uniformly selected from the point cloud. And we use only the coordinates information $P = \{p_i | i = 1, ..., 1024\} \in \mathbb{R}^3$ as the input to the network.

the local geometry. By forming physically explicit geometric relationships in the low-level space, we are able to attach geometric information to the representation of the point cloud. And the richer low-level geometric cues are provided for better implicit geometric feature learning in the subsequent high-level space.

Inspired by the triangular mesh in computer graphics, the KNN method is used to find two nearest neighbors p_{i1},p_{i2} for any sampled point $p_i, i = 1, ..., 1024$ to form a triangle in 3D space, as shown in Fig. 1. The triangular mesh can flexibly present continuous and complex 3D shapes, and then the features of the triangular mesh are used to explicitly enhance the low-level geometric relationships between discrete points and the extracted new geometric descriptor $p_i^{'}$ corresponding to p_i. This geometric descriptor $p_i^{'}$ contains 1)the global position information of point p_i, 2)the vector information from two neighboring points p_{i1} and p_{i2} to point p_i, 3)the side lengths from two points p_{i1} and p_{i2} to point p_i in the triangular mesh, 4)the normals of the triangular mesh, 5)and the center of mass obtained through the intersection of the three medians. The specific formulas are shown below:

$$p_i^{'} = (p_i, edge_1, edge_2, length_1, length_2, normal, centroid); p_i^{'} \in \mathbb{R}^{17} \tag{1}$$

$$p_i = (x_i, y_i, z_i), p_{i1} = (x_{i1}, y_{i1}, z_{i1}), p_{i2} = (x_{i2}, y_{i2}, z_{i2}); p_i, p_{i1}, p_{i2} \in \mathbb{R}^3 \tag{2}$$

$$centroid = (\frac{x_i + x_{i1} + x_{i2}}{3}, \frac{y_i + y_{i1} + y_{i2}}{3}); centroid \in \mathbb{R}^3 \tag{3}$$

$$edge_1 = p_{i1} - p_i, edge_2 = p_{i2} - p_i; edge_1, edge_2 \in \mathbb{R}^3 \tag{4}$$

$$length_1 = |edge_1|, length_2 = |edge_2|; length_1, length_2 \in \mathbb{R}^1 \tag{5}$$

$$normal = edge_1 \times edge_2; normal \in \mathbb{R}^3 \tag{6}$$

$p_i^{'}$ extends the features of p_i from 3 dimensions containing only coordinates to 17 dimensions containing additional geometric information, combining the location and geometric information. The topological structure information between points is enriched. In the point geometry construction module, the obtained new geometric descriptor $p_i^{'}$ is subjected to further feature extraction by MLP to make the feature representation of the point more expressive. The MLP is expressed as a channel fully connected layer. In our experiments, we implement MLP by performing a 1×1 convolution operation on the feature map of the point cloud, as well as batch normalization and activation functions, as shown in the following equations:

$$M(p_i^{'}) := \tau(BN(c_{1\times1}(p_i^{'}))) \tag{7}$$

where M is the MLP,τ is the activation function, BN is the batch normalization, and $c_{1\times1}$ is the convolution, and its subscript indicates the size of the convolution kernel.

3.2 Embeddable Attention Module

The attention system of human brain is to select and to focus on a small portion of useful information from a large amount of input information, then solving the information overload problem by filtering out a large amount of irrelevant information. The proposed embeddable attention module is established based on this mechanism. It enables the neural network to automatically select channel features containing key information for enhancement from a large amount of input information. It helps to improve the ability of data processing in neural network. The limited computational resources will be well utilized for processing more important information and solving the information redundancy problem. To extend its applicability, the module has been designed with the same dimensionalities for both input and output parameters, and it can be directly embedded into other network architectures, as shown in Fig. 1.

The embeddable attention module uses the query-key-value model to map the input features F_1 to three different spaces using different linear transformations to obtain the query matrix Q, the key matrix K and the value matrix V. The linear mapping process is shown below:

$$Q = F_1 W_q \in \mathbb{R}^{D_k \times N} \tag{8}$$

$$K = F_1 W_k \in \mathbb{R}^{D_k \times N} \tag{9}$$

$$V = F_1 W_v \in \mathbb{R}^{D_v \times N} \tag{10}$$

Where W_q,W_k,W_v are shared learnable linear transformations, D_k is the dimension of query matrix and key matrix, D_v is the dimension of value matrix, and

D_k is not necessarily equal to D_v. Firstly, we use the query matrix and the key matrix to calculate the attention weights by matrix dot product:

$$A^{'} = (\alpha^{'})_{i,j} = Q \cdot K^T \tag{11}$$

The attention distribution is obtained by normalizing the weights using the softmax operator and the L_1 parametrization:

$$\alpha_{i,j} = \frac{\alpha^{''}_{i,j}}{\sum_k \alpha^{''}_{i,j}}, \alpha^{''}_{i,j} = softmax(\alpha^{'}_{i,j}) = \frac{exp(\alpha^{'}_{i,j})}{\sum_k exp(\alpha^{'}_{i,j})} \tag{12}$$

Then the weighted sum of the input information is calculated based on the attention distribution and the value matrix to obtain the attention feature F_a:

$$F_a = A \cdot V = (\alpha)_{i,j} \cdot V \tag{13}$$

In order to obtain the output features F_{out} of the embeddable attention module, first of all, the channel attention feature $F^{'}_{in}$ is achieved by mapping, the input feature F_{in} nonlinearly through the fully connected layer fc_1, the activation function ReLU and the fully connected layer fc_2. Then, similar to the Laplace operator, by calculation of element subtraction, the attention feature F_a is replaced by the offset $F^{'}_a$ between the input F_{in} of the attention module and the attention feature F_a. Finally, the matrix sum of $F^{'}_a$ processed by the convolution layer and the channel attention features $F^{'}_{in}$ is the output features. The equations are shown below:

$$F^{'}_{in} = fc_2(ReLU(BN(fc_1(F_{in})))) \tag{14}$$

$$F_{out} = CBR(F^{'}_a) + F^{'}_{in} = CBR(F_{in} - F_a) + F^{'}_{in} \tag{15}$$

In our designed attention module, softmax operator is used in the first dimension and the L_1 parametrization is applied in the second dimension to normalize the attention mapping, which improves the attention weight and reduces the effect of noise. The offset between the input F_{in} of the attention module and the attention feature F_a is also calculated for replacing the attention feature F_a in order to prevent its happening where the absolute coordinates of the same object are completely different under strict transformation. Since the query matrix, key matrix and value matrix are jointly determined by the corresponding linear transformation matrix and the input features $F_{in} \in \mathbb{R}^{N \times d_e}$, they are all order-independent. Moreover, the softmax operator and the weighted sum are both permutation-independent operators. Therefore, the whole attention process is alignment invariant and is well suited for disordered and irregular regions presented by point clouds. The process of nonlinearly mapping of the input features F_{in} to obtain the channel attention features $F^{'}_{in}$ enables our network more adaptive to different channel features and more robust. In our network structure, we also applied stacked EAMs for controlling and optimizing the outputs, based on the global context, fine-grained attention features are generated for the input features and transformed into a high-dimensional feature space which can

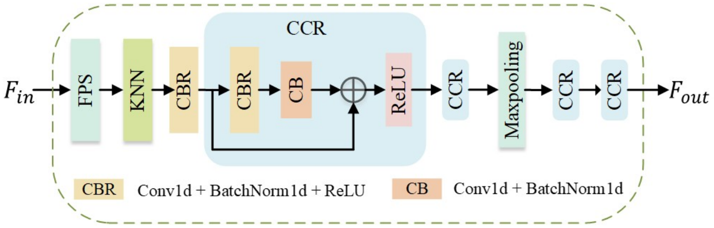

Fig. 2. Residual Connection Pooling Block

characterize the semantic similarity between points, as shown in the following equations:

$$F_1 = EAM^1(F_{in}) \tag{16}$$

$$F_i = EAM^i(F_{i-1}); i = 2, 3, 4 \tag{17}$$

$$F_{final} = MLP(concat(F_1, F_2, F_3, F_4)) \tag{18}$$

3.3 Deep Feature Extraction

As illustrated in Fig. 1, for deep feature extraction, the stacked RMs gradually expand the receptive field and aggregate the local feature information and global relationship information of the point cloud. The RM module initially contains a residual connection pooling block (RCPB). As is shown in Fig. 2, RCPB is a small residual network that improves the information propagation efficiency by adding directly connected edges to the nonlinear convolutional layer, while obtaining a new feature descriptor $p_i^{'}$ for each sampling point p_i. The local geometric context and feature context are fused based on the geometric relation between points. RCPB aggregates the local structure relationships of the point cloud to obtain the local feature F_1. Meanwhile, in order to increase the global relationship information between points, the global relationship matrix G between each point and other points is constructed using the same down-sampling index set Idx_1 as RCPB. Then it is converted into a weight matrix W and multiplied with feature F_1 by elements to get feature F_2. Finally, feature F_1 and feature F_2 are summed by elements to get the output features of RM module.

In the final stage of classification for recognition, in order to extract more representative global shape descriptors of the point cloud, a combination of max pooling and average pooling is applied to obtain the global feature by aggregating the extracted deep features as input. Max pooling is a symmetric function, which is insensitive to the input order and can solve the problem of disorder in point clouds. Also in combination with average pooling, it can avoid ignoring other critical information. To classify point clouds into multiple object classes (e.g., tables, desks, chairs, etc.), we input global features to a classifier. The classifier consists of two cascaded feedforward neural networks with a dropout of 0.5 for each layer, which is finally determined by the linear layer. The class with the highest score in the final predicted classification score is the class of the point cloud.

4 Experiment

We evaluated the performance of our proposed network model on two datasets, ModelNet40 and ScanObjectNN, and compared it with other state-of-the-art methods. The effectiveness and superiority of the modules in this algorithm are demonstrated by ablation experiments.

4.1 Evaluation Indicators

For the evaluation metrics, we use the average accuracy(mAcc) on each category and the overall accuracy(OA) on all categories to evaluate the experimental results, as expressed by the following equations:

$$mAcc = \frac{\sum_{i=1}^{K} \frac{T_i}{N_i}}{K}, OA = \frac{T}{N} \tag{19}$$

where T is the number of correctly predicted point clouds. $T = \sum_{i=1}^{K} T_i$, T_i is the number of correctly predicted point clouds in class i. K is the number of classes in the dataset. N is the number of all point clouds in the dataset.$N = \sum_{i=1}^{K} N_i$, N_i is the number of point clouds in class i.

4.2 Classification on The ModelNet40 Dataset

The ModelNet40 dataset contains 12311 noise-free shape models from 40 classes, 9843 training models and 2468 test models. The experimental batch size is 32, and the initial learning rate is 0.1. The learning rate is adjusted using CosineAnnealingLR. The optimizer is SGD with a weight decay of 0.0002. 300 epochs were trained for the experiment. The experiment was run on GPU 3090 and CPU AMD epyc 7543.

The experimental results are shown in Table 1. Compared with the Transformer-based PointTrans.[20] method, the mAcc improved by 1.1% and OA improved by 0.8%; Compared with CAA, the mAcc improved by 0.7% and OA improved by 0.7%. The mAcc improved by 1.5% and OA improved by 1.6% over the graph convolution-based method DGCNN; The mAcc improved by 3.6% and OA improved by 2.0% over the point convolution-based method PointCNN; The mAcc improved by 1.3% and OA improved by 1.9% over the attention-based method Point2Sequence. Our method obtains higher mAcc value and more competitive OA value, indicating that our method has robust classification performance for different types of point clouds.

4.3 Classification on The ScanObjectNN Dataset

The ScanObjectNN dataset is the first real-world dataset for point cloud classification, containing 15,000 point cloud models in 15 categories. There are 2902 corresponding unique object instances with background, noise and occlusion. The

Table 1. Comparison with the latest methods on the ModelNet40 classification dataset. All cited results are taken from the cited papers. P=points, N=normals. The best is marked in bold.

Method	Input	#Points	mAcc(%)	OA(%)
PointNet [4]	P	1k	86.0	89.2
PointNet++ [5]	P	1k	-	90.7
PointNet++ [5]	P+N	5k	-	91.9
PointCNN [7]	P	1k	88.1	92.5
PCNN [13]	P	1k	-	92.3
PointConv [8]	P+N	1k	-	92.5
Point2Sequence [14]	P	1k	90.4	92.6
RS-CNN [15]	P	1k	-	92.9
DGCNN [10]	P	1k	90.2	92.9
CAA [16]	P	1k	91.0	93.8
PointASNL [18]	P	1k	-	92.9
Point Trans. [17]	P	1k	-	92.8
PosPool [18]	P	5k	-	93.2
MLMSPT [19]	P	1k	-	92.9
PCT [11]	P	1k	-	93.2
Point Trans. [20]	P	1k	90.6	93.7
RepSurf-T [21]	P	1k	91.1	94.0
SGCNN [22]	P	1k	90.4	93.4
3DCTN [23]	P+N	1k	91.2	93.3
ours	P	1k	**91.7**	**94.5**

experimental batch size is 32. The initial learning rate is 0.01. The optimizer is SGD. The weights are decayed by 0.0001, and 200 epochs are trained.

Compared to the current state of the art, our method outperforms all methods with significant improvements in both mAcc and OA. As shown in table 2, our mAcc and OA are 5.5% and 3.8% higher than PRANet, respectively. Furthermore, we note that our method achieves the smallest gap between mAcc and OA. This phenomenon indicates that our method is not biased towards a particular class and shows a fairly good robustness.

4.4 Ablation Studies

The results of the ablation experiments performed on ModelNet40 are shown in Table 3. As can be seen from the first row of data, the removal of the point geometric construction block resulted in a 0.7% decrease in mAcc and a 0.9% decrease in OA. The second row of data shows a 1.3% decrease in mAcc and a 1.4% decrease in OA after removing the embeddable attention module. As can be seen from the third row of data, after removing the global relationship matrix, mAcc decreases by 0.3% and OA decreases by 0.5%. As can be seen from the fourth row of data, using only maximum pooling when performing global feature aggregation, the mAcc decreases by 0.4% and the OA decreases by 0.8%. In summary, each component of the network is effective for point cloud classification recognition.

Table 2. Comparison with the latest methods on the ScanObjectNN classification dataset. All cited results are taken from the cited papers. The best one is marked in bold.

Method	mAcc(%)	OA(%)
PointNet [4]	63.4	68.2
PointNet++ [5]	75.4	77.9
PointCNN [7]	75.1	78.5
SpiderCNN [24]	69.8	73.7
DGCNN [10]	73.6	78.1
BGA-PA++[25]	77.5	80.2
BGA-DGCNN[25]	75.7	79.7
Simple View [26]	-	80.5 ± 0.3
GBNet [16]	77.8	80.5
DRNet [27]	78.0	80.3
PRANet [28]	79.1	82.1
MVTN [29]	-	82.8
RepSurf-T [21]	81.2	84.1
ours	**84.6**	**85.9**

Table 3. Ablation experiments for the proposed network on ModelNet40. GRM indicates global relation matrix.

PGCM	EAM	GRM	Max	Max+Avg	mAcc(%)	OA(%)
✗	✓	✓	✗	✓	91.0	93.6
✓	✗	✓	✗	✓	90.4	93.1
✓	✓	✗	✗	✓	91.4	94.0
✓	✓	✓	✓	✗	91.3	93.7
✓	✓	✓	✗	✓	91.7	94.5

5 Conclusion

In this study, by constructing triangular geometries between points with stacked attention modules focus on the key information, a 3D object recognition algorithm based on point cloud geometry construction and embeddable attention is proposed. To enhance the robustness and effectiveness of the neural network, a combination of max pooling and average pooling is also applied to aggregate global shape descriptors. Due to the advantages of this algorithm, the proposed method has outperformed the most state-of-the-art methods by evaluating ModelNet40 and ScanObjectNN datasets. The proposed method has achieved higher accuracy and has shown good performance in 3D object recognition.

References

1. Chiang, C.H., Kuo, C.H., Lin, C.C., Chiang, H.T.: 3d point cloud clas- sification for autonomous driving via dense-residual fusion network. IEEE Access PP(99), 1–1 (2020)
2. Yang, L., Liu, Y., Peng, J., Liang, Z.: A novel system for off-line 3d seam extraction and path planning based on point cloud segmentation for arc welding robot. Robotics and Computer-Integrated Manufacturing 64(3), 101929 (2020)

3. Bolkas, D., Chiampi, J., Chapman, J., Pavill, V.F.: Creating a virtual reality environment with a fusion of suas and tls point-clouds. International Journal of Image Data Fusion (1), 1–26 (2020)
4. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. IEEE (2017)
5. Qi, C.R., Li, Y., Hao, S., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space (2017)
6. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. IEEE (2020)
7. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: convolution on -transformed points. In: Neural Information Processing Systems (2018)
8. Wu, W., Qi, Z., Li, F.: Pointconv: Deep convolutional networks on 3d point clouds. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
9. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Guibas, L.J.: Kp- conv: Flexible and deformable convolution for point clouds (2019)
10. Yan, S., Yang, Z., Li, H., Guan, L., Kang, H., Hua, G., Huang, Q.: Implicit autoencoder for point cloud self-supervised representation learning. arXiv e-prints (2022)
11. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. 7(2), 13 (2021)
12. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
13. Atzmon, M., Maron, H., Lipman, Y.: Point convolutional neural networks by extension operators. ACM Transactions on Graphics 37(4CD), 71.1– 71.12 (2018) 1
14. Liu, X., Han, Z., Liu, Y.S., Zwicker, M.: Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. Proceedings of the AAAI Conference on Artificial Intelligence 33, 8778–8785 (2019)
15. Fan, B., Pan, C., Xiang, S., Liu, Y.: Relation-shape convolutional neural network for point cloud analysis (2019)
16. Qiu, S., Anwar, S., Barnes, N.: Geometric back-projection network for point cloud classification (2019)
17. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer (2020)
18. Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X.: A closer look at local aggregation operators in point cloud analysis (2020)
19. Han, X.F., Kuang, Y.J., Xiao, G.Q.: Point cloud learning with transformer (2021)
20. Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V.: Point transformer (2020)
21. Ran, H., Liu, J., Wang, C.: Surface representation for point clouds (2022)
22. Liu, S., Liu, D., Chen, C., Xu, C.: Sgcnn for 3d point cloud classification. 2022 14th International Conference on Machine Learning and Computing (ICMLC) (2022)
23. Lu, D., Xie, Q., Xu, L., Li, J.: 3dctn: 3d convolution-transformer network for point cloud classification (2022)
24. Xu, Y., Fan, T., Xu, M., Long, Z., Yu, Q.: Spidercnn: Deep learning on point sets with parameterized convolutional filters (2018)
25. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. IEEE (2020)
26. Goyal, A., Law, H., Liu, B., Newell, A., Deng, J.: Revisiting point cloud shape classification with a simple and effective baseline (2021)

27. Qiu, S., Anwar, S., Barnes, N.: Dense-resolution network for point cloud classification and segmentation (2020)
28. Cheng, S., Chen, X., He, X., Liu, Z., Bai, X.: Pra-net: Point relation- aware network for 3d point cloud analysis. IEEE Transactions on Image Processing PP(99) (2021)
29. Hamdi, A., Giancola, S., Li, B., Thabet, A., Ghanem, B.: Mvtn: Multi- view transformation network for 3d shape recognition (2020)
30. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. IEEE (2017)