ABSTRACT 3D model classification is an important task. Now, 3D model is usually expressed as point cloud. Disorder of point cloud brings great difficulty into 3D model classification. In order to classify 3D model correctly, a new classification method combining Graph Convolution Network (GCN) and Support Vector Machine (SVM) is proposed in this paper. Point cloud is sampled. K-Nearest Neighbor (KNN) algorithm is used to find K nearest points of sampling point, and adjacency matrix is established for graph convolution operation. Shape features \(D_1, D_2, D_3\) and \(A_3\) of sampling point are computed based on its K nearest points. Coordinates and shape features of sampling point are combined as discriminative feature. 2-layer graph convolution is used to aggregate disambiguation information of 1-degree and 2-degree adjacent points of sampling point for describing point cloud comprehensively. At the same time, maximum pooling and average pooling are adopted to retain representative information. Finally, SVM is used to classify point clouds. Experimental results show that compared with GCN based on coordinates, the proposed network improves accuracy of 3D model classification by 1.67%. Global and local information can be extracted adequately when 1024 points are sampled from point cloud. When we select 20 nearest points to compute shape features \(D_1, D_2, D_3, A_3\), local information of point can be described better. Shape features \(D_1, D_2, D_3, A_3\) are combined with coordinates to describe shape and structure of point cloud better. 2-layer graph convolutions are adopted to aggregate information of 1-degree and 2-degree nodes for extracting effective disambiguation features.

INDEX TERMS 3D model, point cloud, graph convolution network, support vector machine, k-nearest neighbor, shape features.

I. INTRODUCTION
3D model classification is an important task in computer graphics and computer vision [1]. It has been applied widely to automatic driving, remote sensing and mapping, 3D reconstruction, intelligent vehicles, forest detection, cultural relic protection and so on. Research on 3D model classification is of great significance. At the beginning, features are extracted manually for 3D model classification. Artificial neural networks are applied to classify 3D model now. Scholars at home and abroad are focusing on 3D model classification. There are mainly 3 solutions for 3D model classification including view-based classification method, voxel-based one, and point cloud-based one.

Rules are adopted to regularize 3D models. In view-based classification method, 3D model is described with 2D views, from which 2D neural networks are applied to extract view features. Then, these view features are merged to represent 3D model [2]. In addition, 2D views can be converted into 3D model in field of 3D reconstruction. It is fundamental for many applications such as object recognition and scene understanding. 2D images are used to infer 3D structure of object so that it can be viewed from all directions. In voxel-based one, voxels are adopted to represent 3D model and 3D CNN (Convolutional Neural Network) is applied to extract features for classifying 3D models [3]. In point cloud-based one, neural networks are directly used to deal with the classification task on point clouds [4].

CNN is suitable for processing regular data, such as images, audio and video. But, it is not suitable to process irregular data. In order to apply convolutional operations to irregular data, researchers put forward GCN [5]. Graph is used to represent objects and their relationships. Convolutional operations on graph are adopted to learn relationships.
between objects. Point cloud is irregular, which has displacement and rotation invariance. So, GCN is used to process point cloud directly for 3D model classification.

Shape descriptor can describe local features of 3D models. $D_1$, $D_2$, $D_3$ and $A_3$ are common shape descriptors. Shape descriptor $D_1$ is Euclidean distance between the centroid and a random point. Shape descriptor $D_2$ is Euclidean distance between two random points. Shape descriptor $D_3$ is the square root of the triangle formed by 3 random points. Shape descriptor $A_3$ is the degree of angle formed by 3 random points.

In this paper, we propose to use GCN and SVM to classify 3D models. 3D model is expressed as point cloud and N points are sampled. KNN algorithm is used to find K nearest neighbors of a point. Its shape feature is computed based on these K nearest neighbors. Coordinates and shape features of points are combined to describe 3D model. Adjacency matrix is established for graph convolution operation. 2-layer graph convolution is used to aggregate disambiguation information of adjacent points of sampling points. Maximum pooling and average pooling are adopted to retain representative information of 3D model. Multi-layer Perceptron (MLP) is applied to extract disambiguation features. MLP is also known as multilayer forward feed neural network, which has excellent nonlinear matching and generalization ability. Back propagation algorithm is used to train MLP to reduce global errors. MLP has characteristics of good nonlinear mapping, high parallelism and global optimization. It is suitable to be introduced into GCN. SVM is adopted to determine 3D model’s category. Modelnet40 is used to testify the proposed network. Experimental results show that the proposed method can classify 3D models effectively. Main contributions of this article are summarized as follows:

- KNN algorithm is used to find K nearest neighbors of a point. Shape feature of a point is computed based on its K nearest points.
- Coordinates $x$, $y$, $z$ and shape features $D_1$, $D_2$, $D_3$ and $A_3$ are combined to describe 3D model.
- 2-layer GCN is used to extract disambiguation information of 3D model and SVM is applied to classify 3D model.

The remainder of this paper is organized as follows. Studies related to our research are summarized in Section II. Discriminative features of 3D model are extracted in Section III. The framework of 3D model classification based on GCN and SVM is described in Section IV. Experiments are provided based on ModelNet40 dataset in Section V. Discussions are described in Section VI. Conclusions are given in Section VII.

II. RELATED WORK

Feature descriptors are used to express shape and structure of 3D model. According to descriptors, 3D model classification is divided into view-based classification method, voxel-based one, and point cloud-based one.

In view-based classification method, 3D model is projected into a series of 2D views from different directions. Feng proposed a group-view CNN for hierarchical correlation modeling towards 3D shape recognition [6]. Wang pooled information from views which were similar and belonged to the same cluster [7]. The pooled features were input to the same layer in a recurrent fashion for boosting the performance of 3D object recognition. Yu aggregated local convolutional features through bilinear pooling to represent 3D object [8]. Bouchou selected suitable snapshots of point cloud and adopted CNN to perform a pixel-wise labeling for each pair of 2D snapshots [9]. Efficient buffering was used to label 3D points for fast back-projection of label predictions in 3D space. Ma proposed a novel multiview-based network for 3D shape recognition and retrieval, which combined CNN with Long Short-Term Memory (LSTM) networks to exploit correlational information of multiple views [10]. When 3D model is reconstructed, useless images are filtered. Preprocess images to reduce noises. Images are detected and matched to find the same feature points. Internal parameters in camera are obtained through calibration operation. Basic matrix and essential matrix are calculated to get 3D spatial positions of feature points [11].

In voxel-based classification method, 3D voxel matrix is used to describe 3D model. Wu used Convolutional Deep Belief network to denote 3D shape as probability distribution of binary variables on 3D voxel grid, which learned the distribution of complex 3D shapes across different categories and arbitrary poses [12]. Riegler presented OctNet which exploited sparsity in the input data and a set of unbalanced octrees to partition 3D space hierarchically [13]. Wang gave an octree-based CNN for 3D shape analysis, which took average normal vectors of 3D model sampled in the closest leaf octants as the input and performed 3D CNN operations on octants occupied by 3D shape surface [14]. Klokov proposed a new deep learning architecture for 3D model recognition which performed multiplicative transformations and shared parameters of these transformations according to the subdivisions of point clouds [15]. Zeng used point clouds as the input and exploited implicit space partition of KD tree to learn local contextual information and aggregate features at different scales [16]. Wang proposed concise multi-scale CNN for point cloud classification, in which local feature and global context were incorporated [17].

In point cloud-based classification method, neural network was used to extract features from point cloud. Qi introduced hierarchical neural network that applied PointNet recursively on a nested partitioning of the input point set, which exploited metric space distances to learn local features with increasing contextual scales [18]. Zhao connected a point with others in local neighborhood to specify its feature based on local region characteristics [19]. At the same time, adaptive feature adjustment was adopted to find the interaction between two points. Lan proposed Geo-CNN which applied generic convolution-like operation to each point and its local neighborhood [20].

In view-based classification method, 3D model is projected into a series of 2D views from different directions. Feng proposed a group-view CNN for hierarchical correlation modeling towards 3D shape recognition [6]. Wang pooled information from views which were similar and belonged to the same cluster [7]. The pooled features were input to the same layer in a recurrent fashion for boosting the performance of 3D object recognition. Yu aggregated local convolutional features through bilinear pooling to represent 3D object [8]. Bouchou selected suitable snapshots of point cloud and adopted CNN to perform a pixel-wise labeling for each pair of 2D snapshots [9]. Efficient buffering was used to label 3D points for fast back-projection of label predictions in 3D space. Ma proposed a novel multiview-based network for 3D shape recognition and retrieval, which combined CNN with Long Short-Term Memory (LSTM) networks to exploit correlational information of multiple views [10]. When 3D model is reconstructed, useless images are filtered. Preprocess images to reduce noises. Images are detected and matched to find the same feature points. Internal parameters in camera are obtained through calibration operation. Basic matrix and essential matrix are calculated to get 3D spatial positions of feature points [11].

In voxel-based classification method, 3D voxel matrix is used to describe 3D model. Wu used Convolutional Deep Belief network to denote 3D shape as probability distribution of binary variables on 3D voxel grid, which learned the distribution of complex 3D shapes across different categories and arbitrary poses [12]. Riegler presented OctNet which exploited sparsity in the input data and a set of unbalanced octrees to partition 3D space hierarchically [13]. Wang gave an octree-based CNN for 3D shape analysis, which took average normal vectors of 3D model sampled in the closest leaf octants as the input and performed 3D CNN operations on octants occupied by 3D shape surface [14]. Klokov proposed a new deep learning architecture for 3D model recognition which performed multiplicative transformations and shared parameters of these transformations according to the subdivisions of point clouds [15]. Zeng used point clouds as the input and exploited implicit space partition of KD tree to learn local contextual information and aggregate features at different scales [16]. Wang proposed concise multi-scale CNN for point cloud classification, in which local feature and global context were incorporated [17].

In point cloud-based classification method, neural network was used to extract features from point cloud. Qi introduced hierarchical neural network that applied PointNet recursively on a nested partitioning of the input point set, which exploited metric space distances to learn local features with increasing contextual scales [18]. Zhao connected a point with others in local neighborhood to specify its feature based on local region characteristics [19]. At the same time, adaptive feature adjustment was adopted to find the interaction between two points. Lan proposed Geo-CNN which applied generic convolution-like operation to each point and its local neighborhood [20].
Local geometric relationships among points were extracted based on edges between the center and its neighbor points. Xu designed SpiderCNN to obtain geometric features from point clouds [21]. SpiderCNN consisted of SpiderConv units which extended convolutional operations from regular grids to irregular point sets. Li presented a permutation invariant architecture for deep learning with orderless point clouds, which used self-organizing map to model spatial distribution of point cloud [22].

With the development of graph convolution, many scholars hope to define general framework of GCN. Bruna considered the generalization of CNN to signals defined in general domains [23]. At the same time, two constructions were given. One was based on hierarchical clustering of the domain, and the other was based on the spectrum of graph Laplacian. Defferrard presented a formulation of CNN, which provided mathematical background and numerical schemes to design convolutional filters on graphs [24]. Kipf gave a semi-supervised learning approach on graph-structured data [25]. It was based on an variant of CNN which operated directly on graphs. Qi proposed 3D graph neural network that built K-nearest neighbor graph of 3D point cloud [26]. Each node corresponded to a set of points and was associated with hidden representation vector. Wang used spectral graph convolution and graph pooling strategy to learn features from point set, in which graph convolution was carried out on a nearest neighbor graph [27].

Graph convolution is divided into spectral method and spatial method. In spectral method, graph convolution is defined on graph and the way of graph convolution is determined in spectral domain. In spatial method, aggregation function is defined to aggregate each central node and its adjacent nodes continuously. Early researchers mainly studied how to define convolution operations on graphs.

In spectral method, Lu constructed a neighborhood graph that reflected relationships between neighbors of each point and used Chebyshev polynomials as graph filters [28]. Li described the first-order GCN in detail, and pointed out that the essence of GCN is Laplace smoothing operation [29]. Xu generalized Weisfeiler-Lehman test to graph convolution, analyzed capabilities of different graph convolution networks, and pointed out that GCN updates features of nodes [30]. They all explain the implementation principle of graph convolution to a certain extent.

In spatial method, hybrid convolution network gave mapping function to solve the lack of translation invariance on graph, which mapped each node’s local structure into vector with the same size, and learned convolution kernel on the mapped result [31]. Message propagation network focused on information propagation and aggregation between nodes, and defined aggregation function to construct the framework [32]. Attention mechanism was considered when aggregation function was selected. Graph attention network (GAT) used attention mechanism to define aggregation function [33]. In GAT, adjacency matrix was used to define relevant nodes. Graph sampling aggregation network (GraphSAGE) was proposed based on GAT, which sampled some adjacent nodes randomly without considering all adjacent ones [34]. Wang proposed neural network EdgeConv to classify and segment point clouds and EdgeConv acted on graphs dynamically [35].

In fact, spectral method can be regarded as a special case and part of spatial method. The difference is that spatial method defines convolution directly in spatial domain, while spectral method maps graph to spectral domain and then defines convolution operation. GCN is a special case of spectral method and the start of spatial method.

In this paper, 3D coordinates and shape descriptors are extracted as disambiguation features from point cloud. GCN and SVM are applied to classify 3D models. The proposed method is described in Figure 1. The proposed method is divided into 3 stages as shown in Figure 1. The first stage is the preprocessing stage, which mainly includes sampling points, calculating 3D shape descriptors and computing Laplace operators. The second stage is neural network calculation stage, which uses GCN to extract features. GCN is trained and testified. The third stage is SVM classification stage, in which features obtained by GCN is input into SVM. SVM is trained and testified. The classification result is given.

![FIGURE 1. The description of the proposed method.](image_url)
used as features of 3D model. Shape functions are easy to be understood and calculated. At the same time, they have good invariance for rigid body rotation and are not sensitive to small disturbances caused by noises. So, shape descriptors are more suitable for processing point cloud. KNN algorithm is used to find K nearest neighbors for sampling point P.

We use KNN to select K nearest neighbor points closest to sampling point. The purpose is to construct the neighborhood space which is composed of these K points. In addition, we take another method to obtain the neighborhood space of sampling point. Sampling point is used as the center and r is taken as radius to draw a sphere. This sphere is used as the neighborhood space and all points in the neighborhood space are neighbors of sampling point. We can use these two methods to acquire the neighborhood space. We use shape descriptors to represent local features of 3D model. Shape descriptors are computed based on the relationship between points in the neighborhood space. However, the second method has two shortcomings. The first one is that point number in the neighborhood space is different for two sampling points. So, it can not ensure that the representation range of shape descriptor is uniform and there is a problem of scale. We should select the same number of neighbor points for all sampling points to compute shape descriptors. The second one is that we need use other method to establish adjacency matrix. When KNN is used to search K nearest points of sampling point, point number in the neighborhood space is the same. There is a strong correlation between sampling point and its neighbor points. The neighborhood space has obvious differences, which ensures that shape descriptors can describe local information better and does not affect the process of establishing adjacency matrix. It omits additional steps and further reduces the expenses of preprocessing.

K nearest points are used to construct the neighborhood space $KPoints_P$ of point P. $KPoints_P$ is shown in formula (2).

$$KPoints_P = KNN(KPoints, P)$$ (2)

where, $KPoints$ is the set including K nearest neighbors of point P.

3D shape descriptors $D1$, $D2$, $D3$, $A3$ are adopted to express shape of $KPoints_P$, which are used to denote shape information of point P. $D1$, $D2$, $D3$, $A3$ of $KPoints_P$ are defined as follows:

$D1$: The distance between a random point in $KPoints_P$ and its centroid $O$.

$D2$: The distance between any two random points in $KPoints_P$.

$D3$: Area of triangle formed by 3 random points in $KPoints_P$.

$A3$: The degree of angle formed by 3 random points in $KPoints_P$.

Centroid $O$ of $KPoints_P$ is computed as follows:

$$O = \frac{1}{|KPoints_P|} \sum_{Q \in KPoints_P} Q$$ (3)

$dist(X, Y)$ between points X, Y is computed as follows:

$$dist(X, Y) = \sqrt{(X - Y)^T(X - Y)}$$ (4)

$D1$ of $KPoints_P$ is defined as follows:

$$D1 = \max(|dist(Q, O)|Q \in KPoints_P))$$ (5)

$D2$ of $KPoints_P$ is defined as follows:

$$D2 = \max(|dist(P_i, P_j)|P_i, P_j \in KPoints_P, P_i \neq P_j)|$$ (6)

We define set $RT$ of combinations of 3 points from $KPoints_P$, which is defined as follows:

$$RT = \{(P_i, P_j, P_k)|P_i, P_j, P_k \in KPoints_P, P_i \neq P_j, P_j \neq P_k\}$$ (7)

Area of triangle $A(P_i, P_j, P_k)$ formed by $P_i$, $P_j$, $P_k$ are computed as follows:

$$A(P_i, P_j, P_k) = \sqrt{s \times (s - a) \times (s - b) \times (s - c)}$$ (8)

where, $s = \frac{(a + b + c)}{2}$, $a = dist(P_i, P_j)$, $b = dist(P_i, P_k)$, $c = dist(P_j, P_k)$.

$D3$ of $KPoints_P$ is defined as follows:

$$D3 = \max(|A(P_i, P_j, P_k)|(P_i, P_j, P_k) \in RT))$$ (9)

Angle R formed by $P_i$, $P_j$, $P_k$ is defined as follows:

$$R(P_i, P_j, P_k) = \frac{b^2 + c^2 - a^2}{2bc}$$ (10)

where, $R$ is angle corresponding to side $P_iP_j$. $A3$ of $KPoints_P$ is defined as follows:

$$A3 = \max(|R(P_i, P_j, P_k)|P_i, P_j, P_k \in RT))$$ (11)

$KPoints_P$ of point P is determined. $D1$, $D2$, $D3$ and $A3$ of $KPoints_P$ are computed as shape features of point P. We combine shape features of P and its coordinates to get discriminative feature $(x, y, z, D1, D2, D3, A3)$. Discriminative feature contains 3D coordinate of point P and its local space information. It makes up for the deficiency that graph convolution can not process local space information. We can aggregate information of each point through graph convolution operation, and gradually obtain discriminative features.
IV. 3D MODEL CLASSIFICATION BASED ON GCN AND SVM

We construct graph as key operator in process of graph convolution. Firstly, we sample N points from point cloud and construct \( N_{Points} \). Secondly, KNN algorithm is used to find K nearest neighbors for each sampling point. Graph \( G = \{V, E, A\} \) is constructed, where \( V \) is a set of N nodes. \( E \) is a set of edges. \( A \) is adjacency matrix, which defines interconnection between nodes. Adjacency matrix \( A_{N \times N} \) is gotten, in which the number of effective elements is \( N \times K \). Distances between sampling point and its K nearest points are respectively computed as correspondent elements of adjacency matrix. Adjacency matrix can characterize relationships between nodes. Adjacency matrix \( A \) is defined as follows:

\[
A_{ij} = \begin{cases} \text{dist}(i, j) & i, j \in N_{Points} \\ 0 & i, j \notin N_{Points} \end{cases}
\]  

Adjacency matrix of point cloud in Figure 3 is shown in Figure 4. KNN is used to find neighbors for each target point of the sampling point. The distance between two points is adopted to express their relationship. 3 target points are taken as examples from Figure 3. Their numbers are respectively 2, 7 and 12. For target point, 7 nearest points are selected as its neighbors. Obviously, 7 neighbors include this target point. Then, they are painted with a color to indicate that these points belong to the same group of neighbors. The edge between two points indicates their distance. Adjacency matrix is used to represent this graph. Number of target point is used as row number of adjacency matrix, and numbers of its neighbor points are used as column number. The distance between target point and its neighbor is viewed as the corresponding element of adjacency matrix. After the above operations are performed on all points, adjacency matrix is gotten which describes relationships between points.

The framework of 3D model classification based on GCN and SVM is shown in Figure 5.

Point cloud is preprocessed. N points are sampled from point cloud. KNN algorithm is used to find K nearest neighbors for sampling point. Shape features \( D_1, D_2, D_3, A_3 \) of point are computed based on its K nearest neighbors. We combine shape features of point and its coordinates to get discriminative feature \( (x, y, z, D_1, D_2, D_3, A_3) \). Adjacency matrix and Laplace matrix are calculated. GCN(7, 64) is used to aggregate information in one degree neighbors of point. MLP(64, 64) is adopted to increase the network’s depth and learn hidden features better. GCN(64, 256) is applied to aggregate information in two degree neighbors. Results of two graph convolutions are concatenated. Then, the result is processed by max pooling and average pooling. After its dimension is decreased by MLP(640, 512, 256, C), the output is gotten. We use MLP to share parameters. Since the output of the first GCN is 64 dimension as shown in Figure 5, the input of the first MLP is 64 dimension. The output of MLP is input to the second GCN. The output of the second GCN is 256 dimension. So, the input of the second MLP is 256 dimension and the dimension of its output is 256. After splicing and pooling operation, the dimension of data becomes 640. So, the input of the third MLP is 640 dimension. It decreases the input data from 640 dimension to 512 dimension. Then, it decreases the dimension of data from 512 to 256. Finally, the data is decreased from 256 dimension to 40 dimension. There are 40 categories of 3D models. We use SVM to determine the category of point cloud.

Graph convolution operations are used to obtain information of its neighbor nodes. Graph convolution operation is shown as follows:

\[
Z = \text{Relu}(LXW) \tag{13}
\]

where, \( W \) represents convolution kernel, \( X \) is the input, \( Z \) is the output, \( \text{Relu} \) is activation function. \( \text{Relu} \) nonlinear activation function is adopted, which is linear rectification function. \( \text{Relu} \) function is shown as follows:

\[
\text{Relu}(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{14}
\]
Relu(x) takes 0 as threshold, which maps more values of feature into 0 and makes it sparse. It avoids the over-fitting problem to a certain extent and is good for feature extraction. Laplace matrix $L$ is shown as follows:

$$L = D^{-1/2}BD^{-1/2}$$  \hspace{1cm} (15)

$$D_{ii} = \sum_{j=1}^{N} B_{ij} \quad i = 1, 2, \ldots, N$$  \hspace{1cm} (16)

where, $B = A + I_N$, $I_N$ is identity matrix. $A$ represents adjacency matrix, which defines the interconnection between nodes. In undirected graph, we have $A_{ij} = A_{ji}$.

The process of classifying point clouds is shown in Figure 6.

![FIGURE 6. The process of classifying point clouds.](image)

The pseudocode of classifying point clouds is shown as follows:

Input: Point cloud $M$
Output: Classification result $C$

1. Point cloud $M$ is sampled to obtain $M1$.
2. Laplace operator $L$ is calculated according to $M1$, which is reserved for graph convolution.
3. Calculate shape descriptors $D1$, $D2$, $D3$ and $A3$ respectively according to $M1$.
4. Shape descriptors $D1$, $D2$, $D3$, $A3$ and $M1$ are spliced together to obtain $V$, and $V$ is input into GCN to obtain the output $O$.
5. $O$ is input into SVM and classification result $C$ is obtained.

A 3D model classification tool based on GCN and SVM is given in this paper. We use GCN and SVM to classify point clouds. Compared with other methods of classifying point clouds, GCN can process point clouds directly without additional operations. It is major advantage for graph convolution to process point clouds directly. In addition, we use 3D shape descriptors to represent local features of point cloud, which are associated with coordinates of point cloud. It strengthens global representation of point cloud. Compared with other methods, the proposed method has advantages. We use SVM to process the output of GCN, which further improves the effect of point cloud classification.

V. EXPERIMENTS

Point cloud is three-dimensional and a collection of points in space. Point cloud used in experiments is ModelNet40 dataset. It can be projected into a series of 2D views from different angles. These 2D views are combined to reconstruct point cloud. In this paper, shape features of point are computed based on its neighborhood space. Shape features and...
coordinates of point are input to GCN to extract disambiguation features. Then, SVM is applied to classify point cloud.

ModelNet40 has 40 categories. There are 9843 models in training set. Test set contains 2840 models. Point cloud is preprocessed before it is input the proposed network. N points are sampled from point cloud. KNN is used to find K nearest points of each sampling point for computing its shape features $D_1$, $D_2$, $D_3$, $A_3$. Adjacency matrix is constructed. Laplace matrix is computed for graph convolution. Shape features of each sampling point and its coordinates are combined as discriminative features.

Five groups of experiments are carried out. The first group of experiments are performed to compare the proposed method, GCN based on shape descriptors and GCN based on coordinates of point cloud. The second group of experiments are conducted to testify the performance of the proposed method respectively under $D_1$, $D_2$, $D_3$ and $A_3$. We compare the proposed method’s accuracy with shape descriptors and without shape descriptors through the second group of experiments. At the same time, we observe the contribution of each shape descriptor to accuracy of 3D model classification. The third group of experiments are performed to testify the influence of sampling point number on the proposed method. Sampling operations are needed in process of classifying point clouds. The number of sampling points also affects the effect of the proposed network to a great extent. So, we set up the third group of experiments to determine optimization sampling number. The fourth group of experiments are conducted to testify the influence of graph convolutional layer number on the proposed network. Each graph convolutional layer learns features from neighbor points of sampling point. If there are more graph convolution layers, points’ features are smoothed excessively which reduces the effect of the proposed network. Therefore, we set up the fourth group of experiments to investigate the influence of the layer number on accuracy of 3D model classification. The fifth group of experiments are performed to testify the influence of the neighbor point number on the proposed method. When KNN is used to select neighbor points of sampling point, the value of K affects adjacency matrix, which influences the effect of GCN. So, we set up the fifth group of experiments to determine optimization value of K.

The first group of experiments include Experiment 1, Experiment 2 and Experiment 3. In Experiment 1, shape descriptors $D_1$, $D_2$, $D_3$, $A_3$ are used as discriminative features and GCN is adopted to classify 3D models. In Experiment 2, coordinates of point cloud are used as discriminative features and GCN is adopted to classify 3D models. In Experiment 3, GCN and SVM are combined to classify 3D models based on shape descriptors $D_1$, $D_2$, $D_3$, $A_3$ and coordinates of point cloud. Test set is adopted to testify accuracies of these 3 experiments as shown in Table 1. Accuracy is defined as shown in formula (17).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

| Categories  | Exp 1 | Exp 2 | Exp 3 |
|-------------|-------|-------|-------|
| airplane    | 0.9700| 0.9900| 0.9900|
| bathtub     | 0.2800| 0.8400| 0.9200|
| bed         | 0.5000| 0.9800| 0.9800|
| bench       | 0.4000| 0.7000| 0.7500|
| bookshelf   | 0.6600| 0.9600| 0.9600|
| bottle      | 0.7600| 0.9400| 0.9500|
| bowl        | 0.1000| 1.0000| 0.8500|
| car         | 0.5700| 0.9500| 0.9600|
| chair       | 0.7200| 0.9700| 0.9700|
| cone        | 0.7000| 0.9500| 0.9500|
| cup         | 0.3500| 0.5000| 0.6500|
| curtain     | 0.4500| 0.8000| 0.8000|
| desk        | 0.2209| 0.8023| 0.8023|
| door        | 0.6000| 0.8500| 0.8000|
| dresser     | 0.4767| 0.7907| 0.8140|
| flower_pot  | 0.2000| 0.0500| 0.3500|
| glass_box   | 0.6800| 0.8400| 0.9600|
| guitar      | 0.9500| 0.9900| 0.9900|
| keyboard    | 0.4500| 0.9500| 1.0000|
| lamp        | 0.4500| 0.8500| 0.7500|
| laptop      | 0.6500| 1.0000| 0.9500|
| mantel      | 0.6200| 0.9300| 0.9400|
| monitor     | 0.5200| 0.9600| 0.9700|
| night_stand | 0.2674| 0.7209| 0.7558|
| person      | 0.4500| 0.8500| 0.8500|
| piano       | 0.5000| 0.8300| 0.8900|
| plant       | 0.5700| 0.8400| 0.8100|
| radio       | 0.2500| 0.7500| 0.7000|
| range_hood  | 0.4600| 0.9100| 0.9100|
| sink        | 0.0500| 0.6000| 0.6000|
| sofa        | 0.5200| 0.9600| 0.9800|
| stairs      | 0.2500| 0.8000| 0.7000|
| stool       | 0.3000| 0.5500| 0.6500|
| table       | 0.2600| 0.7900| 0.8000|
| tent        | 0.4000| 0.9000| 1.0000|
| toilet      | 0.6500| 0.9600| 0.9800|
| tv_stand    | 0.3500| 0.7900| 0.8100|
| vase        | 0.6400| 0.8600| 0.8300|
| wardrobe    | 0.1500| 0.6500| 0.7000|
| xbox        | 0.4500| 0.6500| 0.5500|
| Accuracy    | 0.5348| 0.8728| 0.8874|

TP, FP, FN, TN are shown as follows:

TP(True Positive): Positive samples are predicted as positive categories.

TN(True Negative): Negative samples are predicted as negative categories.

FP(False Positive): Negative samples are predicted as positive categories.

FN(False Negative): Positive samples are predicted as negative categories.

It can be seen from Table 1 that the performance of Experiment 3 is the best. Experiment 3 is better than Experiment 2 at accuracy. This is because that GCN is used to classify 3D models based on coordinates of point cloud in Experiment 2. But, Experiment 3 uses not only shape descriptors but also coordinates of point cloud. At the same
time, GCN and SVM are applied to classify 3D models. This shows that when coordinates of point cloud and shape descriptors are combined, shape and structure of 3D model can be described better. When GCN and SVM are combined, the ability of 3D model classification is increased. So, Experiment 3 is higher than Experiment 2 at accuracy. The performance of Experiment 2 is better than that of Experiment 1. This shows that coordinates have better description ability for 3D model than shape descriptors.

Shape descriptor expresses local information of point cloud and its description ability is influenced by the size of the neighborhood space. Coordinates describe relative position of point in space. If coordinate system changes, coordinate of point will change. When point cloud is rotated, coordinates of points will change. Although point cloud is rotated, it is essentially the same one. This is rotation invariance of point cloud. Shape descriptor can deal with rotation invariance. It represents relative position between points, which is not influenced by coordinates. But, it will be limited by the neighborhood space. Shape descriptor is local feature and ignores global information. Therefore, coordinates have better description ability than shape descriptors. From table 1, we can find that Experiment 2 is better than Experiment 1 at accuracy.

From Table 1, we can see that Experiment 2 achieves better than Experiment 3 under bowl, door, lamp, laptop, plant, radio, stairs, vase, xbox. This is because that models in some categories are similar with those in other ones. For example, some vases are similar to bowls, flowers_pot, cone, and so on. Shape descriptors pay more attention to local information. For two very similar models, global information should be considered when they are classified. But, part of global information will be offset when shape descriptors are introduced into the process of 3D model classification. So, parts of results in Experiment 2 are better than those in Experiment 3.

In order to testify the influence of the proposed network on accuracy of each category, confusion matrix is constructed from Table 1 as shown in Figure 7.

From Figure 7, we can see that a category is incorrectly classified into other categories. For example, category 26 is plant. 19 samples are misclassified and 15 ones are classified into category 15. Category 15 is flower_pot. Category 33 is table. There are 20 instances which are incorrectly classified. 16 ones are classified into category 12. Category 12 is desk. The reason is that there is similarity between plant category and flower_pot category. Desk category and table category is similar with each other. It shows that the proposed method can not distinguish two similar categories well. The similarity between two different categories becomes the biggest factor to improve the performance of classification.

It is multi-classification problem to classify point clouds. AUC(Area Under Curve) is used to testify the proposed method. We calculate false positive case rate (FPR) and true case rate (TPR) for each category under threshold.

From Table 7, we can find that AUC of the proposed network is 0.942.

TPR and FPR are shown as follows:

\[
TPR = \frac{TP}{TP + FN} \quad (18)
\]
\[
FPR = \frac{FP}{FP + TN} \quad (19)
\]

FP, FN, TP, TN of the proposed network is computed based on confusion matrix as shown in Table 2.

From Table 2, we can see that FP of guitar is the smallest and its value is 0. FN of keyboard and tent is the lowest and their values are 0.

In order to verify the effect of shape descriptors, the second group of experiments are conducted. Coordinates of point cloud are denoted as \((x, y, z)\). The proposed method is adopted to classify 3D models respectively based on...
TABLE 2. FP, FN, TP, TN of the proposed network.

| Categories   | FP  | FN  | TP  | TN  |
|--------------|-----|-----|-----|-----|
| airplane     | 1   | 1   | 99  | 2367|
| bathtub      | 2   | 4   | 46  | 2416|
| bed          | 4   | 2   | 98  | 2364|
| bench        | 7   | 5   | 15  | 2441|
| bookshelf    | 15  | 4   | 96  | 2353|
| bottle       | 6   | 5   | 95  | 2362|
| bowl         | 4   | 3   | 17  | 2444|
| car          | 3   | 4   | 96  | 2365|
| chair        | 3   | 3   | 97  | 2365|
| cone         | 1   | 1   | 19  | 2447|
| cup          | 6   | 7   | 13  | 2442|
| curtain      | 6   | 4   | 16  | 2442|
| desk         | 17  | 17  | 69  | 2365|
| door         | 2   | 4   | 16  | 2442|
| dresser      | 19  | 16  | 70  | 2363|
| flower_pot   | 27  | 13  | 7   | 2421|
| glass_box    | 7   | 4   | 96  | 2367|
| guitar       | 0   | 1   | 99  | 2368|
| keyboard     | 1   | 0   | 20  | 2447|
| lamp         | 5   | 5   | 15  | 2443|
| laptop       | 2   | 1   | 19  | 2446|
| mantel       | 5   | 6   | 94  | 2363|
| monitor      | 7   | 3   | 97  | 2361|
| night_stand  | 18  | 21  | 65  | 2364|
| person       | 1   | 3   | 17  | 2447|
| piano        | 4   | 11  | 89  | 2364|
| plant        | 12  | 19  | 81  | 2356|
| radio        | 10  | 6   | 14  | 2438|
| range_hood   | 1   | 9   | 91  | 2364|
| sink         | 2   | 8   | 12  | 2446|
| sofa         | 4   | 2   | 98  | 2364|
| stairs       | 7   | 6   | 14  | 2441|
| stool        | 3   | 7   | 13  | 2445|
| table        | 15  | 20  | 80  | 2353|
| tent         | 9   | 0   | 20  | 2439|
| toilet       | 1   | 2   | 98  | 2367|
| tv_stand     | 9   | 19  | 81  | 2359|
| vase         | 17  | 17  | 83  | 2351|
| wardrobe     | 9   | 6   | 14  | 2439|
| xbox         | 6   | 9   | 11  | 2442|

(x, y, z) in Experiment 4, (x, y, z)+D1 in Experiment 5, (x, y, z)+D2 in Experiment 6, (x, y, z)+D3 in Experiment 7, and (x, y, z)+A3 in Experiment 8. Test set is used to testify accuracies of these 5 experiments as shown in Table 3.

It can be seen from Table 3 that the proposed method achieves the best at accuracy when (x, y, z)+D3 is used as discriminative feature. This shows that discriminative abilities of (x, y, z)+D3 is better than those of (x, y, z)+D1, (x, y, z)+D2, (x, y, z)+A3. The proposed method achieves better at accuracy under (x, y, z) and (x, y, z)+D1 than (x, y, z)+D2 and (x, y, z)+A3. This shows that discriminative abilities of (x, y, z) and (x, y, z)+D1 are better than those of (x, y, z)+D2 and (x, y, z)+A3.

From Table 3, we can see that shape feature D3 achieves the best at accuracy. The reason is that D1 is the distance between any point and the center of point cloud. D2 is the distance between two points. D3 is the area of a triangle. A3 is the angle corresponding to an edge of triangle. The scale of D3 is larger than those of D1, D2 and A3. This is because that D3 describes the relationship between 3 points. But, D1 and D2 only represent the relationship between two points. A3 denotes the relationship between two edges of a triangle. Features with large scale may have more advantages to describe shape and structure of point cloud. Therefore, accuracy of D3 descriptor is the best.

The number of sampling points (S) affects the description of 3D model. When more points are sampled, accuracy of 3D model classification is increased. But, it will improve the scale and complexity of calculation. So, it is necessary to find a balance point. The third group of experiments are performed to investigate the influence of sampling point number on the proposed network. 128, 256, 512, 1024 and
TABLE 4. Accuracies of the third group of experiments.

| Categories    | 128  | 256  | 512  | 1024 | 1280 |
|---------------|------|------|------|------|------|
| airplane      | 1.000| 1.000| 1.000| 0.990| 1.000|
| bathtub       | 0.840| 0.860| 0.880| 0.920| 0.960|
| bed           | 0.940| 0.970| 0.980| 0.980| 0.980|
| bench         | 0.750| 0.750| 0.750| 0.750| 0.750|
| bookshelf     | 0.940| 0.940| 0.940| 0.960| 0.910|
| bottle        | 0.950| 0.940| 0.920| 0.950| 0.920|
| bowl          | 0.950| 1.000| 1.000| 0.850| 0.950|
| car           | 0.940| 0.940| 0.950| 0.960| 0.980|
| chair         | 0.960| 0.970| 0.970| 0.970| 0.960|
| cone          | 0.850| 0.850| 0.850| 0.950| 0.950|
| cup           | 0.450| 0.500| 0.600| 0.650| 0.550|
| curtain       | 0.800| 0.650| 0.900| 0.800| 0.800|
| desk          | 0.825| 0.872| 0.790| 0.803| 0.764|
| door          | 0.950| 0.900| 0.850| 0.800| 0.750|
| dresser       | 0.755| 0.837| 0.825| 0.814| 0.814|
| flower_pot    | 0.100| 0.150| 0.050| 0.350| 1.000|
| glass_box     | 0.830| 0.840| 0.830| 0.960| 0.860|
| guitar        | 0.980| 1.000| 0.990| 0.990| 1.000|
| keyboard      | 0.950| 0.950| 1.000| 1.000| 0.950|
| lamp          | 0.750| 0.850| 0.850| 0.750| 0.800|
| laptop        | 1.000| 1.000| 1.000| 0.950| 1.000|
| mantel        | 0.930| 0.930| 0.940| 0.940| 0.950|
| monitor       | 0.950| 0.930| 0.940| 0.970| 0.940|
| night_stand   | 0.686| 0.697| 0.697| 0.758| 0.720|
| person        | 0.650| 0.750| 0.750| 0.850| 0.830|
| piano         | 0.760| 0.840| 0.890| 0.890| 0.900|
| plant         | 0.650| 0.710| 0.760| 0.810| 0.800|
| radio         | 0.550| 0.600| 0.600| 0.700| 0.650|
| range_hood    | 0.800| 0.880| 0.880| 0.910| 0.930|
| sink          | 0.450| 0.600| 0.550| 0.600| 0.650|
| sofa          | 0.960| 0.960| 0.970| 0.980| 0.970|
| stairs        | 0.650| 0.650| 0.850| 0.700| 0.650|
| stool         | 0.650| 0.650| 0.750| 0.650| 0.800|
| table         | 0.810| 0.750| 0.790| 0.800| 0.810|
| tent          | 0.750| 0.850| 0.900| 1.000| 0.900|
| toilet        | 0.960| 0.950| 0.970| 0.980| 0.960|
| tv_stand      | 0.700| 0.740| 0.730| 0.810| 0.780|
| vase          | 0.800| 0.810| 0.850| 0.830| 0.800|
| wardrobe      | 0.450| 0.450| 0.650| 0.700| 0.650|
| xbox          | 0.600| 0.550| 0.600| 0.550| 0.650|
| Accuracy      | 0.838| 0.856| 0.866| 0.887| 0.874|

1280 points are respectively sampled from point cloud in the third group of experiments. Then, the proposed network is adopted respectively to classify 3D models. Test set is used to testify accuracies of these 5 experiments as shown in Table 4.

It can be seen from Table 4 that as the number of sampling points increases, accuracy of 3D model classification grows. The performance of the proposed method is the best when 1024 points are sampled from point cloud. When 128, 256 and 512 points are sampled, the speed of the proposed network is fast. But, fewer sampled points can not describe shape and structure of 3D model adequately. So, the effect of classification is not ideal. When 1024 points are sampled, more comprehensive description of 3D model is extracted by GCN. The effect of classification is better and the speed of the proposed network is not too slow. When 1280 points are sampled, running time and computation complexity of the proposed network will grow greatly. But, the effect of classification decreases. This is because that noise will be introduced when more points are sampled to describe 3D model.

N points are sampled from point cloud. Disambiguation feature matrix is constructed and its size is N×7. Laplace matrix is constructed and its size is N×N. Graph convolution is the process that disambiguation feature matrix is multiplied by Laplace matrix. So, computation complexity is O(N^2). With the growth of N, computation complexity of the proposed network will increase greatly. Computation complexity is O(128^2), O(256^2), O(512^2), O(1024^2), O(1280^2) when N is respectively set to 128, 256, 512, 1024, 1280.

More points can describe shape and structure of point cloud better. From Table 4, we can see that accuracy of the proposed network grows when the number of sampling points increases. But, noise can also be introduced into disambiguation features. More noises will be introduced when the point number is over 1024. At the same time, accuracy of the proposed network begins to decrease. So, a good balance between feature expression and noises is achieved when 1024 points are sampled.

Graph convolution is a process of aggregating adjacent nodes’ features continuously. Aggregation number affects feature extraction and accuracy of 3D model classification. The fourth group of experiments are conducted to investigate the influence of graph convolutional layer number on 3D model classification. The proposed method is adopted to classify 3D models and convolutional layer number are respectively set to 1, 2, 3 and 4. Test set is used to testify accuracies of these 4 experiments as shown in Table 5.

It can be seen from Table 5 that accuracy of the proposed network first increases and then decreases with the increase of convolutional layer number. When layer number is 2, accuracy of the proposed network is the highest. When one-layer graph convolution is used, one degree nodes’ features are aggregated, which can not describe 3D model completely. So, the effect of 3D model classification is poor. When 3-layer and 4-layer graph convolutions are used, the aggregated features are smoothed excessively and the difference between categories of 3D models is lost. So, accuracy of 3D model classification decreases. When 2-layer graph convolution is used, features of one degree nodes are aggregated. Then, features of two degree nodes are aggregated. This can describe 3D model comprehensively, and the difference between categories of 3D models is not lost. So, the effect of 3D model classification is the best.

KNN is used to extract shape descriptors and establish adjacency matrix. Neighbor point number K affects the performance of 3D model classification. KNN is used to select K nearest points to compute shape features D1, D2, D3 and A3. Shape features D3 and A3 are computed based on 3 neighbor points. Points in the neighborhood space can not be less. So, 5 neighbor points are selected at least. K are respectively...
set to 5, 10, 15, 20, 25, 30 for testifying the influence of K on the proposed network. The fifth group of experiments are performed to testify the influence of K on 3D model classification. The proposed method is adopted to classify 3D models, and K is respectively set to 5, 10, 15, 20, 25, 30. Test set is used to testify accuracies of the proposed network as shown in Table 6.

It can be seen from Table 6 that accuracy of the proposed method is the best when K is set to 20. When K is less than 20, smaller local areas are selected and comprehensive information cannot be extracted from point cloud. Adjacency matrix becomes more sparse and the number of aggregated nodes is less. So, accuracy of 3D model classification is low. When K is greater than 20, larger local area is selected and there is intersection between expressions of different parts in 3D model. There are more redundant information. They interfere with each other and accuracy of 3D model classification is low.

From Table 5 and Table 6, we can see that it is better for the description of sampling point to select 20 neighbor points for computing shape features. At the same time, the proposed network achieves the best at accuracy when convolutional layer number is set to 2. This is because that 20 neighbor points can describe shape and structure of sampling point adequately. At the same time, less noise is introduced. We use the proposed network to process 4 point clouds including bowl, cup, cone and night_stand. Their results in hidden layer are extracted and visualized as shown in Figure 9.
TABLE 7. Accuracies of 5 networks.

| Networks                      | Input points | Model Params/MB | Accuracy |
|-------------------------------|--------------|-----------------|----------|
| VOXNET[3]                     | -            | 0.92            | 0.859    |
| SHAPENETS[11]                 | -            | -               | 0.847    |
| GCN                           | 1024         | 0.63            | 0.873    |
| CNN+LSTM                      | 1024         | 18.14           | 0.829    |
| Ours                          | 1024         | 0.64            | 0.884    |

From Table 7, we can find that accuracy of the proposed network is over other networks and model parameter quantity is lower. The proposed method achieves the best with fewer costs.

VI. DISCUSSIONS

Point cloud is a kind of data in non Euclidean space and irregular. Graph convolution can process data in non Euclidean space. Before point cloud is classified, it need be regularized firstly, such as voxelization or projecting views from many angles for many methods. However, graph convolution can be directly used to process point cloud without regularizing it. At the same time, point cloud has translation and rotation invariance. Calibration operations are needed for many methods. Graph convolution uses graph to represent the relationship between two points, which can deal with these two characteristics of point cloud. So, graph convolution is better than other methods. Graph convolution can aggregate information continuously. When GCN is used to process point cloud, discriminative features are extracted to represent it.

SVM is robust and there is no additional requirement for the dimension of space. So, we use SVM to classify point cloud based on discriminative features. Compared with these existing methods, GCN+SVM can extract more comprehensive features from point cloud efficiently and classify it stably.

Point cloud is a kind of irregular data. It has two characteristics including rotation invariance and disorder. Once point cloud rotates, coordinates of point will change. Rotation invariance means that relative position between points are not influenced by coordinate system. So, point cloud should be regularized after it is rotated. But, it causes information loss to regularize point cloud. GCN can process point cloud directly and it need not be regularized, which avoids information loss. We use shape descriptors to describe local information of point cloud and reduce the influence of rotation operation. Shape descriptors pay more attention to relative position of points in space. Even if points’ coordinates change, shape descriptors will not be affected. So, they can describe point cloud’s characteristics well. In addition, it is relatively simple to calculate shape descriptor and local information can be expressed statistically. Experimental results show that accuracy of the proposed network is improved when graph convolution and shape descriptors are used. We use GCN to extract more comprehensive information from point cloud and SVM is applied to determine its category. Experiments show that the proposed method can improve accuracy of classifying point clouds. At the same time, we find that the proposed network’s parameters affect accuracy of 3D model classification, such as K value, convolutional layer number and the number of sampling points. K value decides the size of neighborhood space, which influences the construction of shape descriptors and Laplace matrix. In the future, we will focus on how to select neighborhood space dynamically, and reduce the influence of K value on shape descriptors and Laplacian matrix as much as possible.

RSCNN is applied to analyze point cloud and uses aggregate functions to aggregate features [36]. There are multiple options for aggregate functions. For GCN, aggregate function is the transformed Laplacian operator. In RSCNN, points in the neighborhood space are selected randomly. RSCNN provides several ways to describe geometric topological relationships in the neighborhood space of the centroid. When Euclidean distance, feature distance and features of point are adopted, it achieves the best. In our method, coordinates of point, Euclidean distance, the distance from point to the centroid, the area of triangle constructed by three points, and the corresponding angle are used. Discriminative features used by RSCNN are related with the way of selecting points from the neighborhood space. RSCNN determines the neighborhood space by the way of dividing spheres. The distribution of points is relatively irregular. It focuses on using Euclidean distance to describe the relationship between points. Our method uses KNN to determine the neighborhood space, which is irregular. But, the distribution of points is relatively regular. So, shape descriptors can be used to describe the relationship between points in the neighborhood space.
We try our best to ensure that the scale of shape descriptors is consistent. DGCNN is applied to classify 3D models based on coordinates of point cloud [35]. DGCNN dynamically constructs graph structure on each layer of the network, takes each point as the central to represent its edge feature with each adjacent point, and then aggregates these features to obtain new representation of the point. Its graph structure can be updated. DGCNN gets local information of 3D model through updating local graph structure. We adopt GCN to classify 3D models based on coordinates, shape descriptors D1, D2, D3, A3 of point cloud. We combine global and local information to describe 3D model. Shape descriptors are used to describe 3D model to keep the invariance of translation, scale and rotation. After GCN constructs graph in coordinate space, graph structure is fixed and will not be updated later.

Generally speaking, multi-view based method needs pre-training and fine tuning in process of extracting features, which causes burden on the whole calculation. Its time complexity is largely determined by the network. Voxel-based method need voxelize data in the preprocessing stage, then data is input into neural network to be processed. This method takes up too much memory in process of calculation, which increases in cube form with the increase of 3D model resolution. The method based on point cloud need organize and construct neural network according to characteristics of point cloud. The network structure is often complex. The calculation complexity is relatively high and the execution time is relatively long. The proposed method in this paper is a method based on point cloud, which uses graph convolution to extract characteristic features from point cloud. Laplacian operator and 3D shape descriptors need be calculated in preprocessing stage. But the network structure is relatively simple, so time complexity of calculation is lower than other methods based on point cloud. Model parameters of the proposed network is only about 0.6 M, and that of PointNet is about 3.4 M. In addition, we use floating point operations, namely FLOPs, to measure the complexity of the proposed network. It is about 589 MFLOPs, and that of PointNet is about 629 MFLOPs. Time complexity of the proposed method is lower than that of PointNet.

VII. CONCLUSION AND FUTURE WORKS

In this paper, 3D model classification network combining GCN and SVM is proposed. It includes 2-layer graph convolution, 3-layer MLP, pooling layer and Support Vector Machine. KNN algorithm is used to find K nearest neighbors of sampling point for building neighborhood space. D1, D2, D3 and A3 of neighborhood space are calculated as shape features of sampling point. Coordinates and D1, D2, D3, A3 are used as disambiguation features. Distances between sampling point and K nearest neighbors are computed, and sparse adjacency matrix is built. Graph convolution is adopted to aggregate disambiguation information of surrounding nodes continuously. Pooling layer is used to retain more representative information. Global features are obtained by MLP. At the same time, SVM is applied to determine 3D model’s category. Several groups of experiments are carried out on ModelNet 40 to compare the performance of the proposed network under various conditions. The number of sampling points is set to 1024. K is set to 20 in KNN algorithm. Shape features D1, D2, D3 and A3 and coordinates are combined as discriminative features. Convolutional layer number is set to 2. Experimental results show that coordinates and D1, D2, D3 and A3 can better describe shape and structure of 3D models. The proposed network can improve accuracy of 3D model classification.

Shape descriptors are used to describe the relationship between each sampling point and its neighbors for expressing local features of 3D model. We combine coordinates and shape descriptors of sampling point as discriminative features to express 3D model adequately. Experimental results show that the combination of coordinates and shape descriptors is effective. It can describe 3D model better and accuracy of 3D model classification is improved. When 1024 points are sampled, accuracy of 3D model classification is the best. It achieves a better balance between sampling noises and the expression of 3D model. In order to ensure the scale uniformity of shape descriptors, we use KNN to find neighbor points for sampling points. Experiments show that when 20 neighbor points are selected, 3D model’s local features can be described better. Through experiments, we find that 2-layer graph convolution can extract more effective features. When the layer number is less, one degree nodes’ features are aggregated and 3D model is not expressed completely. When the layer number is more, the aggregated features are smoothed excessively and the difference between categories of 3D models is lost. When GCN is used to extract discriminative features and SVM is applied to classify 3D models, we find that accuracy of 3D model classification is improved.

In the future, more descriptors will be introduced to express shape and structure of 3D model better. Shape features are computed in dynamic neighborhood space. The proposed method uses KNN to extract K nearest points of sampling point to construct neighborhood space for computing shape features. The number of neighbor points will affect the performance of shape features. It is necessary to compute shape features dynamically in space with different sizes in order to eliminate this influence. If the density of points in neighborhood space is higher, the description ability of shape feature is better. The space with the highest density is selected and shape feature is computed. Therefore, how to obtain shape features from dynamic space will be the focus of future research.

REFERENCES

[1] J. W. H. Tangelder and R. C. Veltkamp, “A survey of content based 3D shape retrieval methods,” Multimedia Tools Appl., vol. 39, no. 3, pp. 441–471, Sep. 2008.

[2] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3D shape recognition,” in Proc. IEEE Int. Conf. Comput. Vis., Santiago, Chile, Dec. 2015, pp. 945–953.
