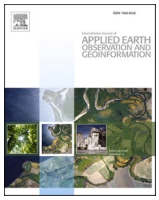
International Journal of Applied Earth Observations and Geoinformation 110 (2022) 102786

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



WGNet: Wider graph convolution networks for 3D point cloud classification with local dilated connecting and context-aware

Yiping Chen [a], Zhipeng Luo [b,*], Wen Li [b], Haojia Lin [b], Abdul Nurunnabi [c], Yaojin Lin [d], Cheng Wang [b], Xiao-Ping Zhang [e], Jonathan Li [b,f]

[a] School of Geospatial Engineering and Science, Sun Yat-sen University, 519082 Zhuhai, China
[b] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China
[c] Department of Geodesy and Geospatial Engineering, Institute of Civil and Environmental Engineering, University of Luxembourg, Luxembourg
[d] School of Computer Science and Engineering, Minnan Normal University, Zhangzhou, China
[e] Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, Canada
[f] Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada

ARTICLE INFO

Keywords:
3D point cloud
Graph convolution networks
3D object classification
Dilated connecting
Context information aware

ABSTRACT

Graph convolution networks (GCNs) have been proven powerful in describing unstructured data. Currently, most of existing GCNs aim on more accuracy by constructing deeper models. However, these methods show limited benefits, and they often suffer from the common drawbacks brought by deep networks, such as large model size, high memory consumption and slow training speed. In this paper, different from these methods, we widen GCNs to improve the descriptiveness by expanding the width of input to avoid the above drawbacks. Specifically, we present a wider GCNs based model, WGNet, for 3D point cloud classification. A local dilated connecting (LDC) module is designed to obtain the adjacency matrix, while a context information aware (CIA) module is proposed to generate initial node representation. These two modules provide a way to transform 3D point cloud into graph structure with larger receptive field and rich initial node features. These two properties widen the channels of input and provide more rich information to describe the samples precisely. Besides, we provide analysis to formulate the above idea as the sample precision description. Then, we adopt ChebyNet as our basic network, and present a skip-connection-based GCNs to improve efficiency of feature reuse. WGNet was evaluated on two datasets. One was acquired by a mobile laser scanning system under the real road environments, while the other was the well-known public artificial dataset, ModelNet40. Experimental results show that WGNet achieves better performance than the state-of-the-art in terms of descriptiveness, efficiency and robustness. Ablation studies also demonstrate the effectiveness of our designed LDC and CIA modules.

1. Introduction

With the advantage of capturing detail 3D point clouds to descript the around environments, LiDAR has been used in many applications, such as autonomous vehicles (AVs) (Chen et al., 2018), intelligent transportation systems (ITS) (Feng et al., 2021; Seo et al., 2015), and Simultaneous Localization and Mapping (SLAM) (Chen et al., 2021; Gong et al., 2019). As a cornerstone of 3D point clouds processing, object classification has important influence on several related tasks, such as sematic and instance segmentation (Hu et al., 2020; Thomas et al., 2019) and object detection (Yang et al., 2020; Qi et al., 2020; Xie et al., 2020). This paper mainly focuses on 3D object classification.

Transitionally, 3D point clouds classification builds on hand-craft-feature descriptors, such as Spin Image (SI) (Johnson and Hebert, 1999), 3D shape context (3DSC) (Frome et al., 2004), and Rotational Projection Statistics (RoPS) (Guo et al., 2013). Recently, deep learning (DL) has been proven as a most powerful technique in 3D point clouds processing. According to the representation, DL based methods can be divided into four classes: volumetric methods (Wu et al., 2014; Maturana and Scherer, 2015), view-based methods (Su et al., 2015), point-set-based methods (Qi et al., 2017), and graph-based methods (Wang et al., 2019). Among those methods, graph-based methods can usually obtain

* Corresponding author.
E-mail addresses: chenyiping@xmu.edu.cn (Y. Chen), zpluo@stu.xmu.edu.cn (Z. Luo), abdul.nurunnabi@uni.lu (A. Nurunnabi), yjlin@mnnu.edu.cn (Y. Lin), xzhang@ryerson.ca (X.-P. Zhang), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2022.102786
Received 25 November 2021; Received in revised form 21 February 2022; Accepted 12 April 2022

better results. The is because graph data structure is match more with the 3D point clouds, and the graph representation can better preserve the distribution pattern of point clouds (Guo et al., 2020). Graph-based methods transform point clouds into graph data, then use graph convolution networks (GCNs) to extract information from graph data, to classify the nodes or graphs. Generally, there are two main classes: methods in spectral domain and spatial domain.

Methods in spectral domain are generally build on the well-developed graph theory. They can be considered as a spectral filtering method. As a pioneering work, ChebyNet (Defferrard et al., 2016) is proposed to approximate the spectral filtering by using the truncated Chebyshev polynomial method. ChebyNet tries to solve the challenges brought by the high dimension and irregular data. Besides, RGCNN (Te et al., 2018) presents a convolution operation oriented to regular graph structure. To solve the challenge brought by the difference of topological structure, Li et al., (2018) proposed AGCN based on SGC-LL module. It uses a learnable distance to measure the similarity between points in a graph. Besides, the spectral CNN model proposed by Bruna et al., (2014) extends the classical convolutional neural network architecture to the graph data processing neighborhood, so that the extended framework can be applied to spectrum processing. In view of the difficulty of spectral CNN model in obtaining better pooling effect, Yi et al., (2017) introduced the idea of hole convolution commonly used in the field of image processing into spectral data processing, and designed a SyncSpecCNN model based on spectral transformation. Several methods are also proposed to extract features by operating on the local graph, such as LocalSpecGCN (Wang et al., 2018), PointGCN (Zhang et al., 2018) and 3DTI-Net (Pan et al., 2019).

Graph-based methods in spatial domain takes operations in spatial way. They can be considered as improved versions of PointNet (Qi et al., 2017). ECCNet (Simonovsky and Komodakis, 2017) is the pioneering work that constructs each point as a vertex of the graph. Build on ECCNet, DGCNN (Wang et al., 2019) constructs a dynamic graph to extract local features at multiple levels. Furthermore, LDGCNN (Zhang et al., 2019a) removes the transformation network in DGCNN, and connects the multi-level features learned by different layers. Several DGCNN based methods have been proposed, such as the DPAM (Liu et al., 2019), KCNet (Shen et al., 2018), G3D (Dominguez et al., 2018), and A-CNN (Komarichev et al., 2019). Though these graph-based methods have achieved superior performance, they suffer from the common drawbacks brought by deep networks, such as the larger model size, higher memory consumption, and slower training speed (Rong et al., 2019; Li et al., 2019). Consequently, it is natural that that seeking the possibility of widening the neural networks to avoid these problems.

As pointing out in Lu et al., (2017), a wide network can have a positive impact on the performance of the model in some cases. In addition, in Zhang et al., (2019b), the authors observe that directly stacking deeper networks cannot bring significant improvements. Besides, Shang et al., (2016) widens the networks by proposing an effective activation scheme, called CReLU, to improve the utilization efficiency of feature maps in each layer. In DenseNet (Huang et al., 2017), it has been proved to be very useful to compensate the characteristics of the current channel by the information of other channels. However, these methods mainly focus on image field, and applying this basic idea on 3D point cloud processing is rarely paid attention to.

In this paper, we widen GCNs by expanding the width of input, rather than stacking deeper networks, for 3D point clouds classification. We propose our model, named WGNet, by designing two modules, the local dilated connecting (LDC) and context information aware (CIA) modules. More specifically, motivated by dilated convolution in image processing, the LDC module is presented to construct the edges of graph with expanding the receptive field. For node signal, we propose the CIA module based on context information, including the distribution characteristics of neighborhood points and local dimension features. Finally, built on the above modules, we adopt ChebyNet (Defferrard et al., 2016) as our basic network, and present a skip-connection-based GCNs to improve efficiency of feature reuse.

Expensive experiments on the public simulated ModelNet40 (Wu et al., 2014) and a real dataset acquired by an MLS system in real road environments, demonstrate the effectiveness of the proposed method. Furthermore, we investigate the efficiency and robustness of WGNet. Additionally, several ablation studies demonstrate the effectiveness of the design wider networks. There are three main contributions.

1) We present a local dilated connecting (LDC) module to generate the adjacency matrix for a graph, which would expand the receptive field of graph convolution to encode more information.

2) We design a context information aware (CIA) module to extract the node features as the initial input of GCNs. Different from normal initial input with just only coordinates, for each point, CIA module embeds the distribution characteristics of its neighborhood points and its local dimension features, resulting in rich distribution pattern awareness.

3) We construct skip-connection-based GCNs to maximize the reuse of features learned by each level. Combined with LDC and CIA modules, a wider and efficient GCNs is proposed to mine richer features to compensate the insufficiency on the depth of GCNs.

This paper is organized as follows. Section 2 describes the proposed WGNet. Section 3 and Section 4 presented and discussed the experimental results, respectively. Section 5 concludes the paper.

2. The proposed method: WGNet

2.1. Over view

We develop a wider GCNs based method for 3D point clouds classification, named WGNet. As presented in Fig. 1, the architecture of WGNet consists of three main parts: an edge structure building module based on local dilated connecting (LDC), a node feature generation module based on context information aware (CIA), and a GCNs based framework with skip-connection. Several symbols are shown in Table 1.

In the following subsections, we will first formulate our model as a model regularization task from the view of loss function. Then, we present the details for the LDC and CIA modules. Finally, built on LDC and CIA modules, we develop our model with a skip-connection GCN.

2.2. Preliminaries

In this paper, we adopt the ChebyNet as our basic network. Since the original graph generated from 3D point cloud is a directed graph and the adjacency matrix $\boldsymbol{A}$ is asymmetric, which cannot meet the requirement, we take a simple way to transform $\boldsymbol{A}$ as a symmetric matrix: $\boldsymbol{A}^{sym} = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T)$. This transformation can be interpreted that the average values of the in- and out- degrees of the graph nodes are regarded as new adjacency relations, that is, the original adjacency matrix $\boldsymbol{A}$ only reflects the out-degree, while in $\boldsymbol{A}^{sym}$, the in- and out- degrees are all considered. This can be regarded as "erasing" the direction of the edge to be an undirected one. Therefore, the above transformation does not change the graph structure.

In GCN, the generalized formulation of convolved result, $\boldsymbol{Z}^{(l+1)}$ in l-th layer, can be written as:

$$\boldsymbol{Z}^{(l+1)} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{Z}^{l}\Theta^{l} \tag{1}$$

where $\boldsymbol{Z}^{l}$ is the convolved result in (l-1)-th layer, $\widetilde{\boldsymbol{A}} = \boldsymbol{A}^{sym} + \boldsymbol{I}_n$ means adding self-connections to the adjacency matrix,$\boldsymbol{I}_n$ is $n \times n$ identity matrix,$\widetilde{\boldsymbol{D}}$ is degree matrix with $\widetilde{\boldsymbol{D}}_{i,j} = \sum_j \widetilde{\boldsymbol{A}}_{i,j}$, and Θ^{l} is the matrix of filter parameters in l-th layer. We denote the final output of convolution layers as $\boldsymbol{Z}^{L}$. Then the loss function for graph classification is:

$$\mathscr{L} = -\sum_{j=1}^{C_{cat}} Y_j \ln\left(softmax\left(\boldsymbol{Z}^{(L)}\right)_j\right) \tag{2}$$

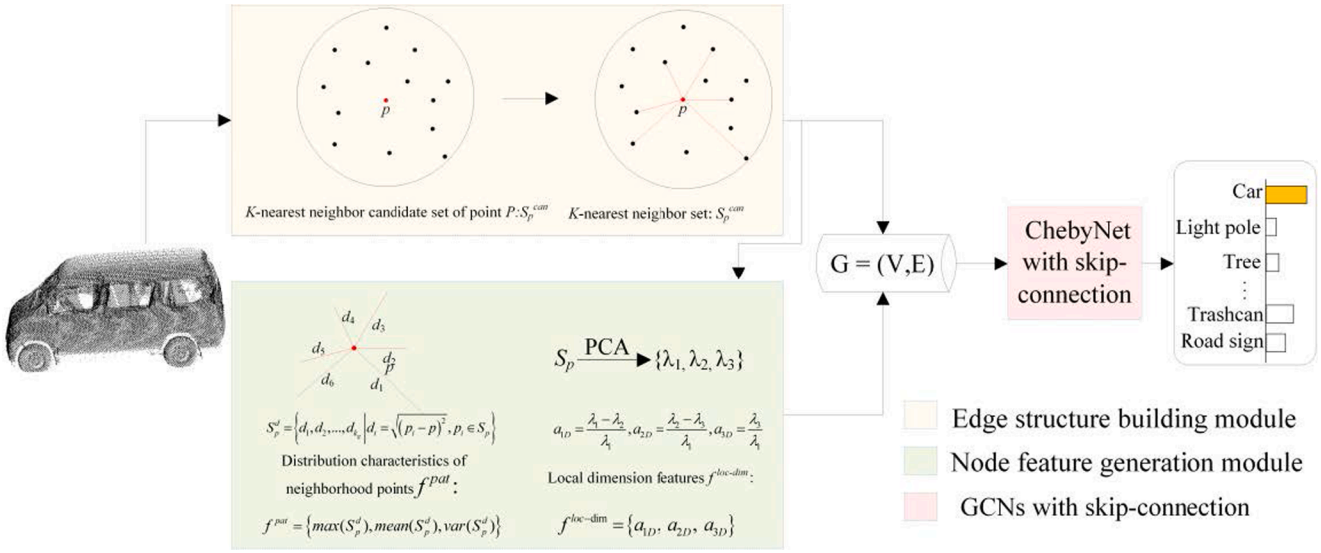


Fig. 1. Architecture of WGNet, which consists of three main parts: edge structure building, node feature generation and GCNs with skip-connection.

Table 1
Several important symbols used in this paper.

Symbols	Description
$P = \{p_i \in R^3 \mid i = 1, \cdots, n\}$	3D point clouds with n points
$\boldsymbol{p}_i = (x_{p_i}, y_{p_i}, z_{p_i})$	3D coordinates of $\boldsymbol{p}_i$
$\mathscr{V} = \{v_i\}_{i=1}^n$, $v_i = (\boldsymbol{p}_i, \text{Other properties})$	Node set of the graph
$\boldsymbol{X}_{n\times 3}$, $\boldsymbol{X}(i,:) = \boldsymbol{p}_i$	Matrix form of node set
$\mathscr{E} = \{\mathscr{E}_{i,j} \mid i,j = 1, \cdots, n\}$, $\mathscr{E}_{i,j}$=1 if v_i and v_j are connected	Edge set of the graph
$\boldsymbol{A}$, $\boldsymbol{A}_{i,j} = \mathscr{E}_{i,j}$	Adjacency matrix
$\boldsymbol{D} = \text{diag}(d_1, \cdots, d_n)$, $d_i = \sum_j \boldsymbol{A}_{i,j}$	Degree matrix
$\mathscr{G} = (\mathscr{V}, \mathscr{E})$	A graph data representation

where C_{cat} is the number of categories. $\mathbf{Y} = \{Y_1, \cdots, Y_{C_{cat}}\}^T$ is the ground truth with one-hot form. We investigate the loss function. It can be further written as.

$$\mathscr{L} = -\boldsymbol{Y}^T\boldsymbol{Z}^{(L)} + \ln\left(\sum_{j=1}^{C_{cat}} e^{Z_j^{(L)}}\right)\boldsymbol{Y}^T\mathbf{I} \tag{3}$$

where $\mathbf{I}$ is a column vector with $\mathbf{I}(k,1) = 1, \forall k \in \{1, \cdots, C_{cat}\}$. Now we take an insight in $\boldsymbol{Z}^{(L)}$. Firstly, we denote N_{knn} as the number of neighbor points, and each point has the same number of neighbor points. Therefore, $\widetilde{\boldsymbol{D}} = (N_{knn}+1) \bullet \boldsymbol{I}_n$. Then, according to (1), we have.

$$\boldsymbol{Z}^{(l+1)} = \frac{\widetilde{\boldsymbol{A}}^2}{(N_{knn}+1)^2}\boldsymbol{Z}^{(l-1)}\Theta^{(l-1)}\Theta^l \tag{4}$$

With $\boldsymbol{Z}^{(0)} = \mathbf{X}$,$\boldsymbol{Z}^{(L)}$ is:

$$\boldsymbol{Z}^{(L)} = \frac{\widetilde{\boldsymbol{A}}^L}{(N_{knn}+1)^L}\boldsymbol{X}\Theta \tag{5}$$

where $\Theta \triangleq \Theta^{(L-1)}\Theta^{(L-2)}\cdots\Theta^{(0)}$. Combining (3) and (5), the loss function is:

$$\mathscr{L} = -\boldsymbol{Y}^T\frac{\widetilde{\boldsymbol{A}}^L}{(N_{knn}+1)^L}\boldsymbol{X}\Theta + \ln\left(\sum_{j=1}^{C_{cat}} e^{\left(\frac{\widetilde{\boldsymbol{A}}^L}{(N_{knn}+1)^L}\boldsymbol{X}\Theta\right)_j}\right)\boldsymbol{Y}^T\mathbf{I} \tag{6}$$

As shown in (6), $\mathscr{L}$ is determined by three parts: the adjacency matrix $\widetilde{\boldsymbol{A}}$, the node features $\boldsymbol{X}$ and the weights Θ learned from networks. Since a graph data is represented by $\widetilde{\boldsymbol{A}}$ and $\boldsymbol{X}$, we can mine the spatial distribution and geometric structure between neighbor points to describe the input samples more precisely. Following this idea, in this work, we design two modules, the LDC and CIA, to impose constraint on adjacency matrix and node features, respectively. We will introduce the details for the LDC and CIA module in following subsections.

2.3. Local dilated connecting (LDC) module

As the common way to build the adjacency relation, standard kNN method still suffers from two main issues, the sensitivity to noise and imbalance between receptive field and efficiency. Therefore, to convert discrete 3D point cloud into graph data, it is worth to consider an edge construction method with better robustness and wider receptive field. Inspired by dilated convolution (Yu and Koltun, 2016; Chen et al., 2017), which is widely used in image processing, we adopt a similar idea and design a local expansion connection strategy.

Specifically, as shown in Fig. 1,$\forall \mathbf{p}_i \in P$, standard kNN is firstly used to obtain the k-nearest neighbor candidate set with $2N_{knn}$ points:

$$\mathrm{S}_{\mathbf{p}_i}^{\text{cand}} = \{\mathbf{p}_1, \cdots, \mathbf{p}_{2N_{knn}}\} \tag{7}$$

Then, the nearest neighbor points are selected from the candidate set. Generally, the selection methods can be divided into random selection and fixed position selection. To improve the robustness, we adopt random selection to obtain the neighbor set with N_{knn} points:

$$\mathrm{S}_{\mathbf{p}_i} = \left\{\mathbf{p}_1, \cdots, \mathbf{p}_{N_{knn}} \mid \mathbf{p}_k \in \mathrm{S}_{\mathbf{p}_i}^{\text{cand}}, k = 1, \cdots, N_{knn}\right\} \tag{8}$$

Repeating the above operation for each point, we can construct the new adjacency matrix $\boldsymbol{A}^{\text{dila}}$.

We now analyze LDC module from the view of sample description. Firstly, $\boldsymbol{A}^{\text{dila}}$ can be represented as:

$$\boldsymbol{A}^{\text{dila}} = \boldsymbol{A} - \boldsymbol{A}^- + \boldsymbol{A}^+ \tag{9}$$

where $\boldsymbol{A}_{i,j}^- = 1$ if the edge $\mathscr{E}_{i,j}$ is erased (i.e.,$\mathbf{p}_j \in \mathrm{S}_{\mathbf{p}_i}^{\text{cand}}$ and $\mathbf{p}_j \notin \mathrm{S}_{\mathbf{p}_i}$), else $\boldsymbol{A}_{i,j}^- = 0$, and $\boldsymbol{A}_{i,j}^+ = 1$ if $\mathscr{E}_{i,j}$ is added (i.e.,$\mathbf{p}_j \in \mathrm{S}_{\mathbf{p}_i}^{\text{cand}}$ and $\mathbf{p}_j \in \mathrm{S}_{\mathbf{p}_i}$), else $\boldsymbol{A}_{i,j}^+ = 0$. Then,

$$\widetilde{\boldsymbol{A}}^{\text{dila}} = \left(\frac{\boldsymbol{A}^{\text{dila}} + (\boldsymbol{A}^{\text{dila}})^T}{2}\right) + \mathbf{I}_n = \widetilde{\boldsymbol{A}} + \Delta_A \tag{10}$$

where $\Delta_A = \frac{\boldsymbol{A}^+ + (\boldsymbol{A}^+)^T - \boldsymbol{A}^- - (\boldsymbol{A}^-)^T}{2}$. Then,

$$(\widetilde{\boldsymbol{A}}^{\text{dila}})^L = (\widetilde{\boldsymbol{A}} + \Delta_A)^L = \binom{L}{0}\widetilde{\boldsymbol{A}}^L + \binom{L}{1}\widetilde{\boldsymbol{A}}^{L-1}\Delta_A + \cdots + \binom{L}{L}(\Delta_A)^L \triangleq \widetilde{\boldsymbol{A}}^L + \Phi_A \tag{11}$$

where $\Phi_A = \binom{L}{1}\widetilde{\boldsymbol{A}}^{L-1}\Delta_A + \cdots + (\Delta_A)^L$. Then, according to (5), the new result is.

$$\boldsymbol{Z}^{(L)}_{new} = \frac{(\widetilde{\boldsymbol{A}}^{\text{dila}})^L}{(N_{knn}+1)^L}\boldsymbol{X}\Theta = \frac{\widetilde{\boldsymbol{A}}^L + \Phi_A}{(N_{knn}+1)^L}\boldsymbol{X}\Theta \triangleq \boldsymbol{Z}^{(L)} + \mathscr{U} \tag{12}$$

where $\mathscr{U} = \frac{\Phi_A}{(N_{knn}+1)^L}\boldsymbol{X}\Theta$. Now we take an insight in $\mathscr{U}$, which can be rewritten as.

$$\mathscr{U} = \frac{\Phi_A\boldsymbol{X}\Theta}{(N_{knn}+1)^L} = \frac{\boldsymbol{I}_n\Phi_A\boldsymbol{X}\Theta}{(N_{knn}+1)^L} = \frac{\boldsymbol{I}_n \bullet (\Phi_A\boldsymbol{X}) \bullet \Theta}{(N_{knn}+1)^L} = \frac{(\boldsymbol{I}_n)^L}{(N_{knn}+1)^L} \bullet (\Phi_A\boldsymbol{X}) \bullet \Theta \tag{13}$$

We can view $\mathscr{U}$ as the result of a new input. More specifically, the adjacency matrix of this new input is $\boldsymbol{I}_n$, which means nodes on this graph are all isolated. Besides, the node set of this new input is the same as that of the raw input, however, the signal for each node has new representation. From (13), we can observe that the matrix form of the new graph is $\Phi_A\boldsymbol{X}$. Combing (12) and (13), the function of LDC module can be considered as adding an extra sample as input. This extra sample shares the same model parameters Θ and the result is fused into $\boldsymbol{Z}^{(L)}$ to obtain the final output.

2.4. Context information aware (CIA) module

Node information is an important part of graph data. 3D coordinates are usually selected as the node signal (Qi et al., 2017; Wang et al., 2019). However, 3D point clouds are not isolated. Instead, its geometric structure and spatial information are determined by the local neighborhood point-distribution pattern. Therefore, only considering the coordinate will reduce the representation ability. Built on this consideration, we design a context information aware (CIA) module to generate the node representation. As shown in Fig. 1, it consists of two kinds of features, the distribution characteristics of neighborhood points and local dimension features.

2.4.1. Distribution characteristics

Distribution characteristics are described with three important statistical characteristics of local neighborhood points. Firstly, as shown in Fig. 1, $\forall \mathbf{p}_i \in P$, the distance metric set is formed by the distance between $\mathbf{p}_i$ and each k-nearest neighbor point:

$$S^d_{\mathbf{p}_i} = \{d_i | d_i = |\mathbf{p}_j - \mathbf{p}_i|, \mathbf{p}_j \in S_{\mathbf{p}_i}\} \tag{14}$$

where k-nearest neighbor set, $S_{\mathbf{p}_i}$, is obtained by (9). Then, several special statistical measures are selected to form the distribution characteristics. The simplest way is to take the above all distances as the distribution characteristics directly. However, in real environments, the robustness will be greatly affected due to the unavoidable presence of outliers and uneven density. To balance the descriptiveness and robustness, three statistical measures are utilized: the maximum, mean and variance of the distance set. Therefore, the distribution characteristics, is defined as follows:

$$f^{pat}_{\mathbf{p}_i} = \{\max\left(S^d_{\mathbf{p}_i}\right), mean\left(S^d_{\mathbf{p}_i}\right), var(S^d_{\mathbf{p}_i})\} \tag{15}$$

Note that, $f^{pat}_{\mathbf{p}_i}$ can not only maintain robustness, but also describe the distribution information. The maximum value describes the radius of the neighborhood. More specifically, when the number of neighborhood points is fixed, the larger the maximum value, the larger the neighborhood radius, which means the point may be in a position with lower density. Besides, average value shows the average distance between point $\mathbf{p}_i$ and its local neighbors. It reflects the contribution of each neighborhood point to $\mathbf{p}_i$, as well as the point density in the local space. In addition, the variance describes the degree of uniformity of among the neighboring points. When the maximum and average values are fixed, a greater variance means that the distribution of neighborhood points relative to $\mathbf{p}_i$ is more discrete. Therefore, the point may be in the position where the shape changes greatly, such as the edge and the corner of space.

2.4.2. Local dimension feature

To further enhance the points information, we introduce dimensional features (Gressin et al., 2013; Lin et al., 2014) to describe the local geometric shape. For an arbitrary point $\mathbf{p}_i$, we first obtain the set of adjacent points, $S_{\mathbf{p}_i}$, from (8). Then, using the principal component analysis (PCA), we can calculate three eigenvalues, λ_1, λ_2, and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). Finally, the local dimension feature can be calculated as follows:

$$f^{loc-dim}_{\mathbf{p}_i} = \{a_{1D}, a_{2D}, a_{3D}\} \tag{16}$$

where, $a_{1D} = \frac{\lambda_1-\lambda_2}{\lambda_1}$, $a_{2D} = \frac{\lambda_2-\lambda_3}{\lambda_1}$, $a_{3D} = \frac{\lambda_3}{\lambda_1}$. Note that, $f^{loc-dim}_{\mathbf{p}_i}$ describes the basic local geometric structure. As shown in Fig. 2(a), when λ_1 is far greater than λ_2 and λ_3, a_{1D} is far greater than both a_{2D} and a_{3D}, and the local point clouds present an approximate linear structure. In Fig. 2(b), when λ_1 and λ_2 are close, and far greater than λ_3, a_{2D} is much larger than a_{1D} and a_{3D}, and the local point cloud enjoys an approximate planar structure. When three eigenvalues are approximately equal, a_{3D} is greater than a_{1D} and a_{2D}, the local point cloud tends to be discrete, as shown in Fig. 2(c).

Therefore, $f^{pat}_{\mathbf{p}_i}$ and $f^{loc-dim}_{\mathbf{p}_i}$ provide additional and useful information to describe the structure of point clouds. Combined with coordinates, they constitute more richer node information. That is, for each point $\mathbf{p}_i$ on graph $\mathscr{G}$, the node information is given as follows:

$$f^{\mathscr{G}}_{\mathbf{p}_i} = \{x_{\boldsymbol{p}_i}, y_{\boldsymbol{p}_i}, z_{\boldsymbol{p}_i}, f^{pat}_{\mathbf{p}_i}, f^{loc-dim}_{\mathbf{p}_i}\} \tag{17}$$

We now analyze the effect of CIA module. Obviously, the adjacency matrix $\boldsymbol{A}$ and diagonal matrix $\boldsymbol{D}$ remain unchanged, while $\boldsymbol{X}'$ has new representation, i.e.,

$$\boldsymbol{X}' = \boldsymbol{X} \bullet [\boldsymbol{I}_3, \boldsymbol{O}_{3\times 6}] + [\boldsymbol{O}_{n\times 6}, \Delta^{pat}_X, \Delta^{loc-dim}_X] \tag{18}$$

where $\boldsymbol{I}_3$ is an identity matrix, $\boldsymbol{O}_{3\times 6}$ and $\boldsymbol{O}_{n\times 6}$ are zero matrices. Δ^{pat}_X and $\Delta^{loc-dim}_X$ are 3×3 matrices, determined by $f^{pat}_{\mathbf{p}_i}$ and $f^{loc-dim}_{\mathbf{p}_i}$, respectively, i.e., $\Delta^{pat}_X(i,:) = f^{pat}_{\mathbf{p}_i}$ and $\Delta^{loc-dim}_X = f^{loc-dim}_{\mathbf{p}_i}$, $\forall i = 1, \cdots, n$. Hence, according to (5), the new result is.

$$\boldsymbol{Z}^{(L)}_{new} = \frac{\widetilde{\boldsymbol{A}}^L}{(N_{knn}+1)^L}\boldsymbol{X}'\Theta = \widetilde{\boldsymbol{A}}^L\frac{\boldsymbol{X} \bullet [\boldsymbol{I}_3, \boldsymbol{O}_{3\times 6}]}{(N_{knn}+1)^L}\Theta + \widetilde{\boldsymbol{A}}^L\frac{[\boldsymbol{O}_{n\times 6}, \Delta^{pat}_X, \Delta^{loc-dim}_X]}{(N_{knn}+1)^L}\Theta \tag{19}$$

Since the result of $\widetilde{\boldsymbol{A}}^L\frac{\boldsymbol{X} \bullet [\boldsymbol{I}_3, \boldsymbol{O}_{3\times 6}]}{(N_{knn}+1)^L}\Theta$ is equivalent to that of $\boldsymbol{Z}^{(L)}$ in (5), we just need to analyze the second item, which is denoted as $\mathscr{V}$, i.e.,

$$\mathscr{V} = \frac{\widetilde{\boldsymbol{A}}^L}{(N_{knn}+1)^L}[\boldsymbol{O}_{n\times 6}, \Delta^{pat}_X, \Delta^{loc-dim}_X]\Theta \tag{20}$$

Similarly, $\mathscr{V}$ can be considered as an extra new input, and its adjacency matrix is $\widetilde{\boldsymbol{A}}$, which means it has the same graph structure as the raw input. Besides, from (20) we can observe that the new input has new representation. Its matrix form is $[\boldsymbol{O}_{n\times 6}, \Delta^{pat}_X, \Delta^{loc-dim}_X]$, which is determined by $f^{pat}_{\mathbf{p}_i}$ and $f^{loc-dim}_{\mathbf{p}_i}$. Therefore, the function of CIA module can be viewed as adding an extra sample as input, which considers rich local context information. This extra sample shares the same graph structure and model parameters, and the result is fused into $\boldsymbol{Z}^{(L)}$ to obtain the final output.

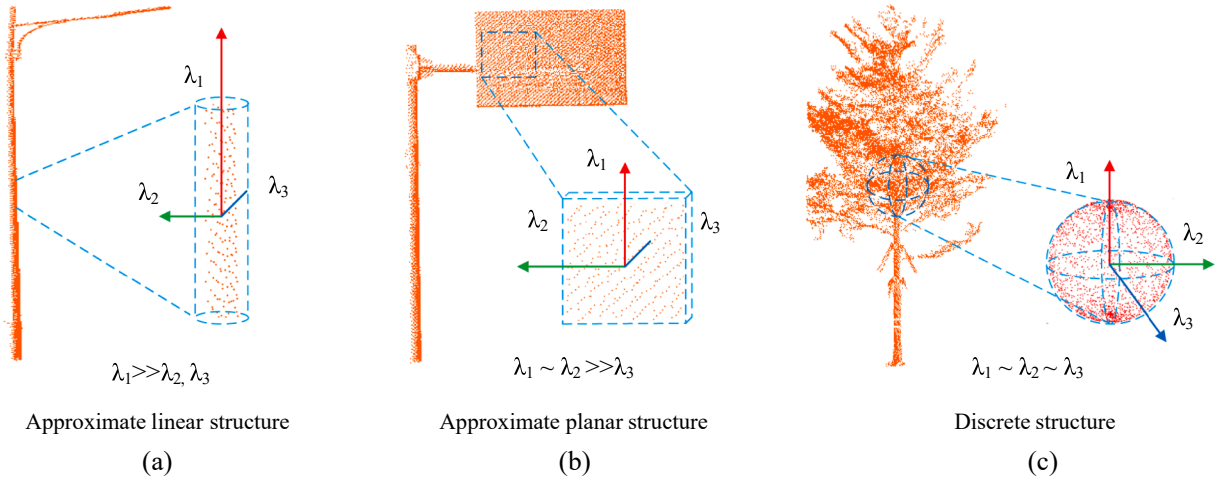


Fig. 2. The diagram of local dimension features. The relationship between three eigenvalues reflects the spatial distribution of local point clouds: (a) when λ_1 is far greater than λ_2 and λ_3, it presents an approximate linear structure; (b) when λ_1 and λ_2 are closer and far greater than λ_3, it presents an approximate planar structure; and (c) when the three eigenvalues are almost similar, it presents a discrete/volumetric structure.

2.5. Wider GCNs with skip-connection

As shown in Fig. 3, WGNet takes ChebyNet (Defferrard et al., 2016) as the basic network. After operating LDC and CIA modules, we can obtain a wider graph data, $\mathscr{G}^{'} = (\mathscr{V}^{'}, \mathscr{E}^{'})$, where the adjacency matrix $\widetilde{\boldsymbol{A}}^{\text{dila}}$ and node initial feature matrix $\boldsymbol{X}^{'}$ are given by (11) and (19) respectively. Therefore, the input signal is a tensor with shape of $N \times 9$.

The framework consists of two branches. The upper branch takes the initial graph data as input, and utilizes a multi-layer perceptron (MLP). The parameter, $\{a_1, a_2, \ldots, a_n\}$, is the number of neurons in MLP and a_i is the number of neurons in i^{th} layer. The output of MLP,f^{up}, is then sent to each ChebyNet layer to concatenate the output of convolution. Note that, the output of MLP tends to contain richer shallow features, which is of great benefit to make up for the features produced by networks with the lack of depth.

The lower branch consists of several basic ChebyNet modules, following by batch normalization (BN) and ReLU activation function. A global max pooling layer and fully connected layer are attached with the graph convolution layer. The output of full connection layer is a $C_{cat} \times 1$ tensor, denoted as $\boldsymbol{Z}^{'}$, which is taken as the final probability score vector. Then, the loss function for WGNet is:

$$\mathscr{L} = -\sum_{k=1}^{C_{cat}} Y_k \ln(softmax(\boldsymbol{Z}^{'})_k) \tag{21}$$

Note that, since LDC and CIA are pluggable modules, it can be applied in another basic network. In experiments, we apply other backbones to validate the generalization and effectiveness of them.

3. Experiments

This section evaluates the performance of WGNet. Section 3.1 introduces the dataset and experimental platform. Section 3.2 obtains the optimal combination of parameters. Section 3.3 evaluates the performance of WGNet on the classification task. Section 3.4 and 3.5 analyze the efficiency and robustness, respectively. In Section 3.6, ablation experiments are conducted to verify the effectiveness of LDC and CIA.

3.1. Dataset description and implementation

Two datasets, Sub-RObject10 and ModelNet40, are used to evaluate WGNet. Sub-RObject10 were collected by a mobile laser scanning system in real road environments. It consists of 1676/401 3D common road objects from 10 categories in train/test samples. ModelNet40 (Wu et al., 2014) is a well-known public dataset with 40 kinds of 3D shapes. Note that, since in training stage, the Laplacian matrices of the graphs of training samples would be read into memory, there is strict restriction on the number of graphs and nodes, and large-scale datasets are difficult to be used in this work. Therefore, we randomly divided ModleNet40 into four sub-sets, named Subset1, 2, 3 and 4, respectively. Table 2 and 3 list the details.

WGNet was implemented on a desktop computer with Ubuntu 16.04 system, Inter (R) Xeon(R) E5-2678 v3 2.50 GHz, GeForce RTX 3090. We trained our method with Pytorch 1.4.0. The initial learning rate is 0.001, and was reduced by half every 20 epochs. The number of epochs is 120, and the batch size is 8. The maximum number of graph nodes of each sample is 256.

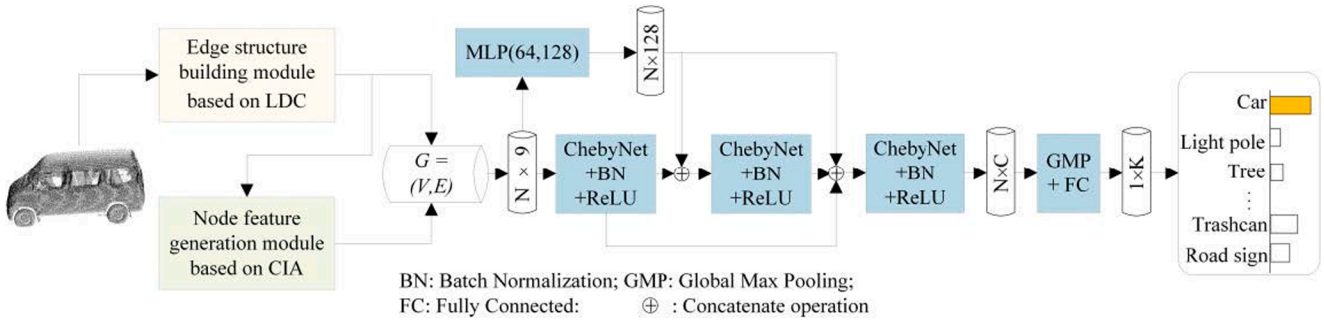


Fig. 3. The networks of WGNet. It consists of three main parts: edge structure building module, node feature generation module, and GCNs with skip-connection. ChebyNet is used as the basic framework.

Table 2
Details of Sub-RObject10.

	Bus station	car	Light-pole	pedestrian	Road sign
Number of training samples	112	184	232	126	128
Number of testing samples	28	42	56	24	34
Category ID	1 Station sign	2 Traffic light	3 Traffic sign	4 Trashcan	5 Tree
Number of training samples	105	120	234	144	291
Number of testing samples	24	24	59	38	72
Category ID	6	7	8	9	10

Table 3
Details of four Sub-dataset of ModelNet40.

Dataset	The category ID in ModelNet40
Subset1	1, 2, 3, 13, 14, 15, 25, 26, 27, 37
Subset2	4, 5, 6, 16, 17, 18, 28, 29, 30, 38
Subset3	7, 8, 9, 19, 20, 21, 31, 32, 33, 39
Subset4	10, 11, 12, 22, 23, 24, 34, 35, 36, 40

3.2. WGNet generation parameter setting

WGNet has three important parameters: the number of convolutional layers L, the number of neighborhood points N_{knn}, and the parameter K_{cheby} in Chebyshev polynomials. These three parameters have different effects on the model. Specifically, L affects the ability to capture the information distribution on graph, while N_{knn} and K_{cheby} affect the receptive field of the convolution. Generally, a larger N_{knn} will bring a larger receptive field of convolution, and the larger the K_{cheby} is, the higher the order of the adjacency relationship considered in feature extraction, which enhances the receptive field of the model. To obtain the best combination, we tested the influence of different parameters on Sub-RObject10. PR (precision-recall) curve is used as the metric. The ranges of L, N_{knn}, and K_{cheby} are set to {2, 3, 4, 5}, {10, 15, 20} and {1, 2, 3} respectively. Grid search is used to obtain optimal combination. Additionally, the network structures and models' names corresponding to different convolutional layers are shown in Table 4.

3.2.1. The number of convolution layers l_G

Fig. 4 shows the PR (precision-recall) curves of WGNet with different parameter settings. Each row in the figure shows the PR curves of WGNet with different K_{cheby} and same N_{knn}. With the same K_{cheby} and N_{knn}, the PR curves of different network structures are different, which shows the network structure affects the performance of WGNet. From the first row of Fig. 4, it can be observed that the curves of WGNet(2) (read curves) and WGNet(3) (green curves) are located in the upper position, which shows these two networks achieve good results. In addition, it can be observed from the second row that WGNet(2) and WGNet(3) obtain better performance than other network structures under the same parameters. Moreover, the third row shows a similar performance. In summary, WGNet(2) and WGNet(3) perform well in most cases. Although WGNet(5_2) also achieves excellent results in some cases, it obtains relatively poor performance with other parameter settings. The main reason for the above results is that in graph convolution network, as the number of model layers' increases, the problem of gradient disappearance will get worse, resulting in over-smooth in the back propagation, which limits the model's ability to describe features. Therefore, the structures of WGNet(2) and WGNet(3) are taken as the frameworks in the proposed method.

Table 4
Configurations of different network structures.

Number of convolution layer	Kernels in each convolution	Name
2	(64)- (1024)	WGNet(2)
3	(64)- (64)- (1024)	WGNet(3)
4	(64)- (64)- (128)- (1024)	WGNet(4)
5	(64)- (64)- (64)- (128)- (1024)	WGNet(5_1)
5	(64)- (128)- (256)- (512)- (1024)	WGNet(5_2)

3.2.2. The number of neighborhood points N_{knn}

By comparing different columns, it can be observed that as the number of neighborhood points increases, the performance improves. For example, for the first column, as the number of neighborhood points increases, the overall recall significantly improves. Specially, the lowest recall in Fig. 4(g) is less than 97%, while the lowest recall in Fig. 4(d) is 97% and that in Fig. 4(a) is 97.5%. The second and third columns of Fig. 4 show similar situations. To analyze the impact of the number of neighborhood points in detail, we fixed the network structure and set $K_{cheby} = 2$, and evaluated the performance of WGNet with different N_{knn}. Fig. 5 shows the PR curves. As shown in Fig. 5, as the number of neighborhood points increases from 10 to 20, the position of PR curve of each network structure model becomes higher. This result indicates that the performance of WGNet has improved. Specially, as shown in Fig. 5 (a), as the number of neighborhood points increases, the PR curve of WGNet(2) rises significantly, which proves that the number of neighborhood points has great impact on WGNet. The reason for these results is that a large N_{knn} can expand the receptive field of the model, so that the convolution kernel can obtain more neighborhood information, thereby effectively improving performance. However, the increase in the number of neighborhood points will rapidly increase the computational complexity, then affect the efficiency of the model. Therefore, to balance classification performance and calculation efficiency, we set $N_{knn}N_{knn} = 20$.

3.2.3. Convolution kernel parameters K_{cheby}

Each column in Fig. 4 shows the PR curves of the models corresponding to different neighborhood points and network structures with the same K_{cheby}. By comparing with the different columns, it can be observed that most of the PR curves in the second column have a higher position, while that in the first and third columns are relatively low. This indicates that WGNet can obtain the best performance with $K_{cheby} = 2$. To analyze the effect of parameter K_{cheby} K_{cheby} on WGNet in detail, we fixed the network structure and the number of neighborhood points, and evaluated the performance of WGNet with different K_{cheby}. Fig. 6 shows the results. As shown in Fig. 6(a) and (b), it can be observed that WGNet (2) and WGNet(3) have the highest curve position with $K_{cheby} = 2$. These results show that these two network structures achieve the best performance with $K_{cheby} = 2$. As shown in Fig. 6(c), WGNet(5_2) performs best with $K_{cheby} = 3$. In summary, the parameter K_{cheby} determines order of the node adjacency used in the graph convolution. The higher the value, the larger the adjacency order considered during convolution and the larger receptive field. However, a large receptive field may cause the model to be affected by noise greatly, and the accuracy of the local information depicted will also be reduced. Therefore, K_{cheby} is set to 2, which means the second- order neighborhood information of the node is considered during convolution.

3.3. Descriptiveness analysis

3.3.1. Experiments on sub-RObject10

To evaluate the descriptiveness of WGNet under real environments, we conducted experiments on Sub-RObject10 and compared WGNet with two existing methods, JointNet (Luo, et. al., 2019) and RSSNet

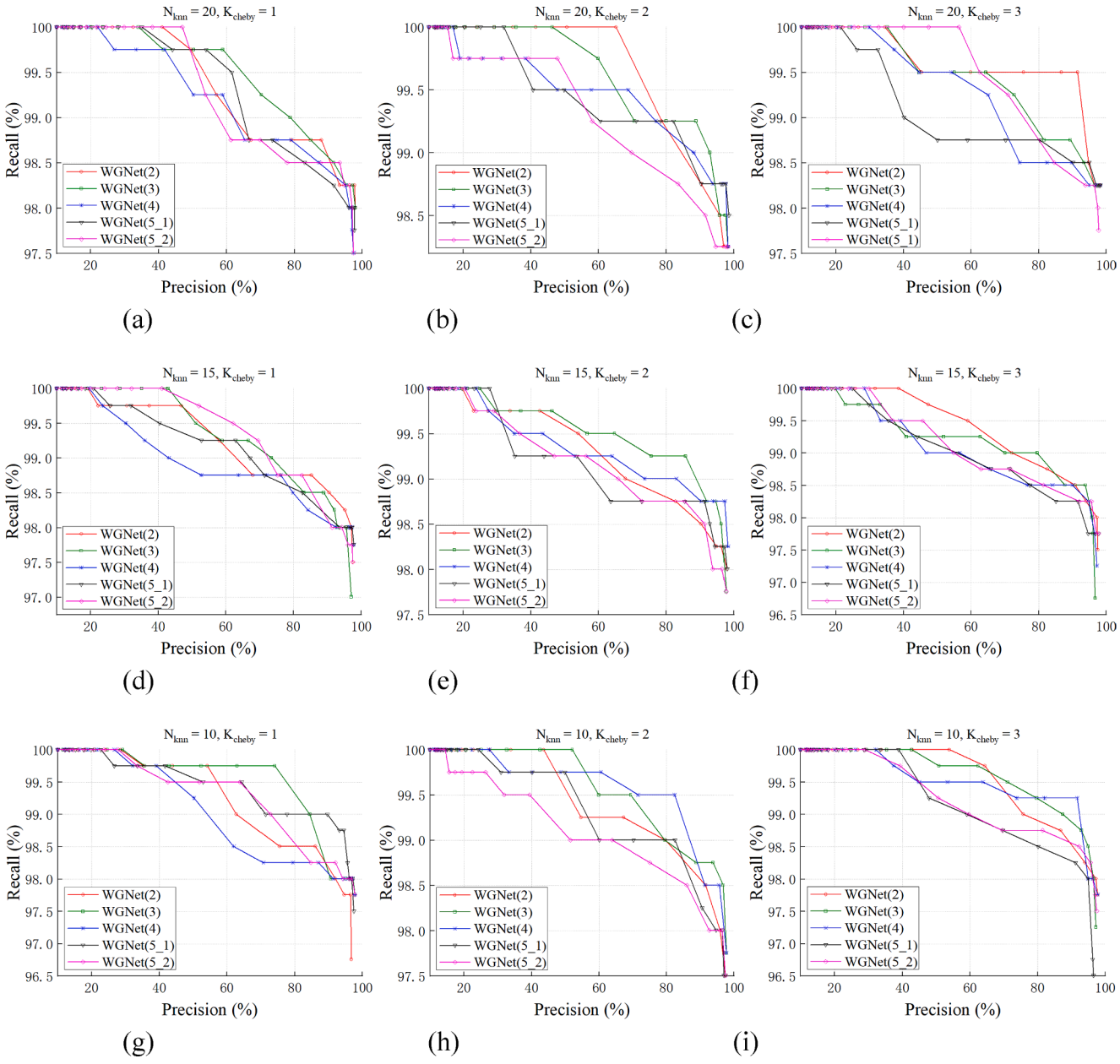


Fig. 4. PR curves of WGNet with different parameter settings.

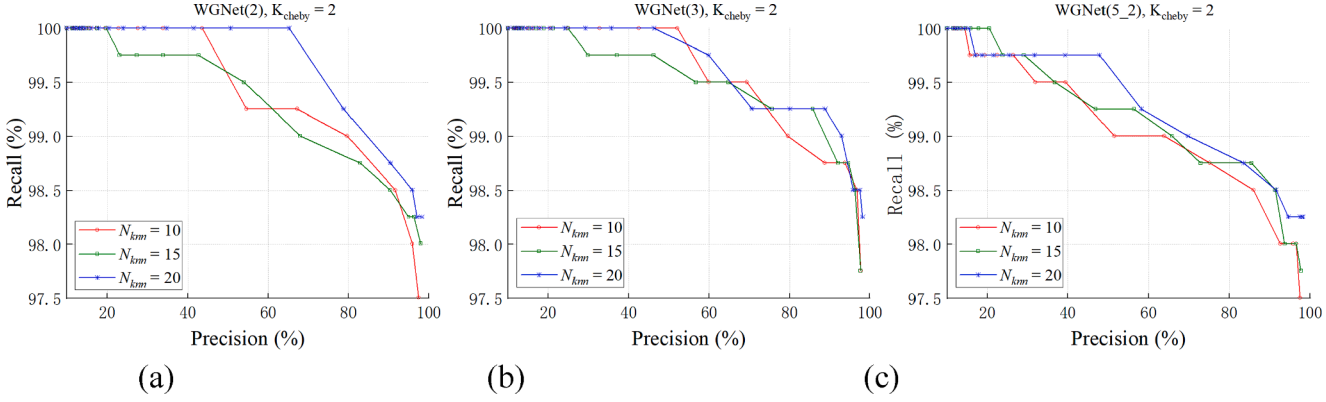


Fig. 5. PR curves of WGNet with different values of N_{knn} (fixed the network structure and set $K_{cheby} = 2$).

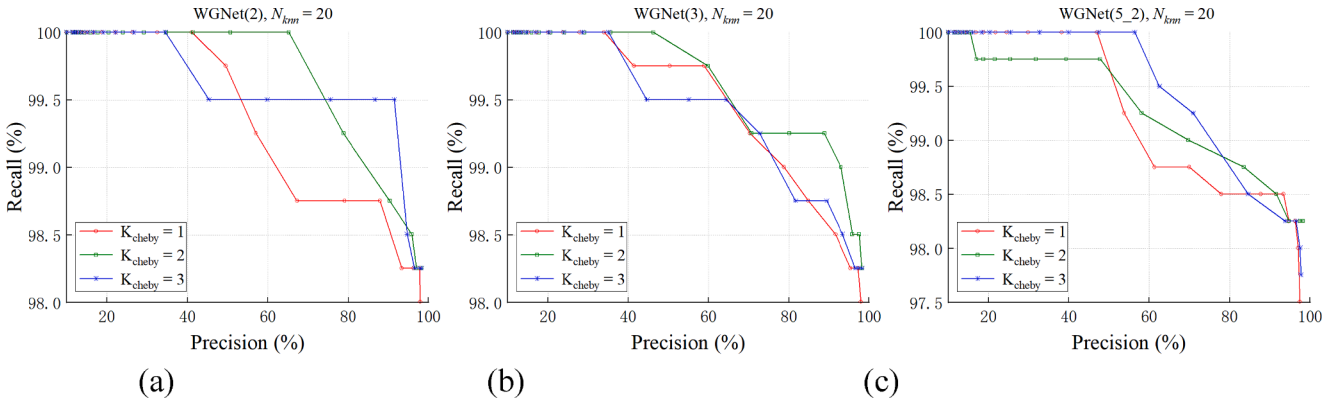


Fig. 6. PR curves of WGNet with different values of..K_{cheby}

(Luo, et. al., 2020). The metrics are mean accuracy (MA) and overall accuracy (OA).

Table 5 shows the comparison results. It can be observed that WGNet (3) obtains the best results on both MA and OA. More specifically, WGNet(3) outperforms JointNet and RSSNet with improvements of 7.39% and 0.9% in terms of MA, and about 8% and 1% in terms of OA, respectively. WGNet(2) also yields competitive results. However, compared to the WGNet with two graph convolutional layers, i.e., WGNet(2), the advantage of WGNet(3) is not obvious. In addition, Fig. 7 presents the visualization results learned by different models using t-SNE.

3.3.2. Experiments on ModelNet40

We tested the performance of WGNet on the four subsets of ModelNet40. Two representative models, PointNet (Qi et. al., 2017) and DGCNN (Wang et. al., 2019), were selected as comparison methods. Table 6 gives the results. Observely, WGNet achieves the best results on four subsets. WGNet(3) outperforms other methods in both of MA and OA on subset 1, 2 and 4, while WGNet(2) achieves the best performance in both two metrics on subset 3. Specially, on subset 4, the MA of WGNet (3) is 6.9% and 8.2% higher than that of PointNet and DGCNN, respectively. Fig. 8 presents the visualization results on subset1.

In summary, WGNet has stronger descriptiveness than other methods. More specifically, compared with DGCNN, the advantage of our method mainly comes from the larger receptive fields and the richer information of nodes. Firstly, for DGCNN, the receptive field of each node is determined by the standard kNN method, while WGNet adopts the designed LDC strategy, which would expand the size of receptive fields. Secondly, in DGCNN, 3D coordinates are directly regarded as the information of nodes. Different from DGCNN, the node information of WGNet contains not only the coordinate values, but also the context information, including the distribution characteristics of neighborhood points and the local shape dimension features. Therefore, although WGNet does not update the graph in feature space, the two proposed modules can compensate the information loss caused by without updating the graph, and enhance the descriptiveness. Figuratively, WGNet uses the widen input information to compensate the deficiency of "shallow" model.

Table 5
The recognition performance comparison on Sub-RObject10 (in %).

Metric	JointNet	RSSNet	WGNet(2)	WGNet(3)
MA	90.71	97.20	97.92	**98.10**
OA	91.25	97.26	**98.25**	**98.25**

3.4. Efficiency analysis

3.4.1. Space and time complexity of the model

Space complexity is usually evaluated by model size and GPU-memory consumption, and the time of forward-backward is used to evaluate the time complexity. The comparison methods include two graph-based methods: PointNet (Qi et. al., 2017) and DGCNN (Wang et. al., 2019). Table 7 reports the results.

For space complexity, the model size of WGNet is smaller than that of DGCNN. The reason is that WGNet has a small number of convolution layers and so the number of model parameters is less than DGCNN. In addition, comparing with PointNet, WGNet(2) has obvious advantage over model size. Besides, we can observe that the size of WGNet increases approximately linearly with the increase of parameter K_{cheby}. This is because the increase of K_{cheby} will enlarge the receptive field of graph convolution, thus increasing the number of training parameters. As to PointNet, because it uses point-based representation and the sizes of most of convolution kernels are 1×1, the memory consumption is the smallest. Compared with point-based representation, in a graph structure input, the neighborhood points need to consume additional memory. Therefore, the memory consumption for WGNet is slightly larger than PointNet.

For time complexity, WGNet has efficient training and testing performance. Especially, the backpropagation time of WGNet(2) with $K_{cheby} = 2$ is about 4 times quicker than that of DGCNN. This high time efficiency of the proposed method is mainly due to the shallow network structure and smaller convolution kernel size. These two properties reduce the computational burden of gradient and the number of parameters to be updated. In addition, since PointNet uses point-based representation and the small sizes of convolution kernels, it shows significant advantage over other methods.

3.4.2. Efficiency of LDC module

As discussed in Section 2, the random selection is adopted to choose the neighbor points in LDC module, which may bring additional time consumption when comparing with the standard kNN strategy. To evaluate the effect of random selection on the efficiency of the LDC module, we conduct experiments for different neighbor point selection strategies. Fig. 9 shows the compared results. Obviously, the time consumption of the LDC strategy is very close to that of the standard kNN strategy. More specifically, the maximum differences of time consumption between these two strategies are 0.0047 s, 0.0524 s, and 0.0826 s when the numbers of points are 1,000, 5,000 and 10,000, respectively. These results demonstrate that in LDC module, searching more neighbor points has very little effect on the efficiency of the LDC module.

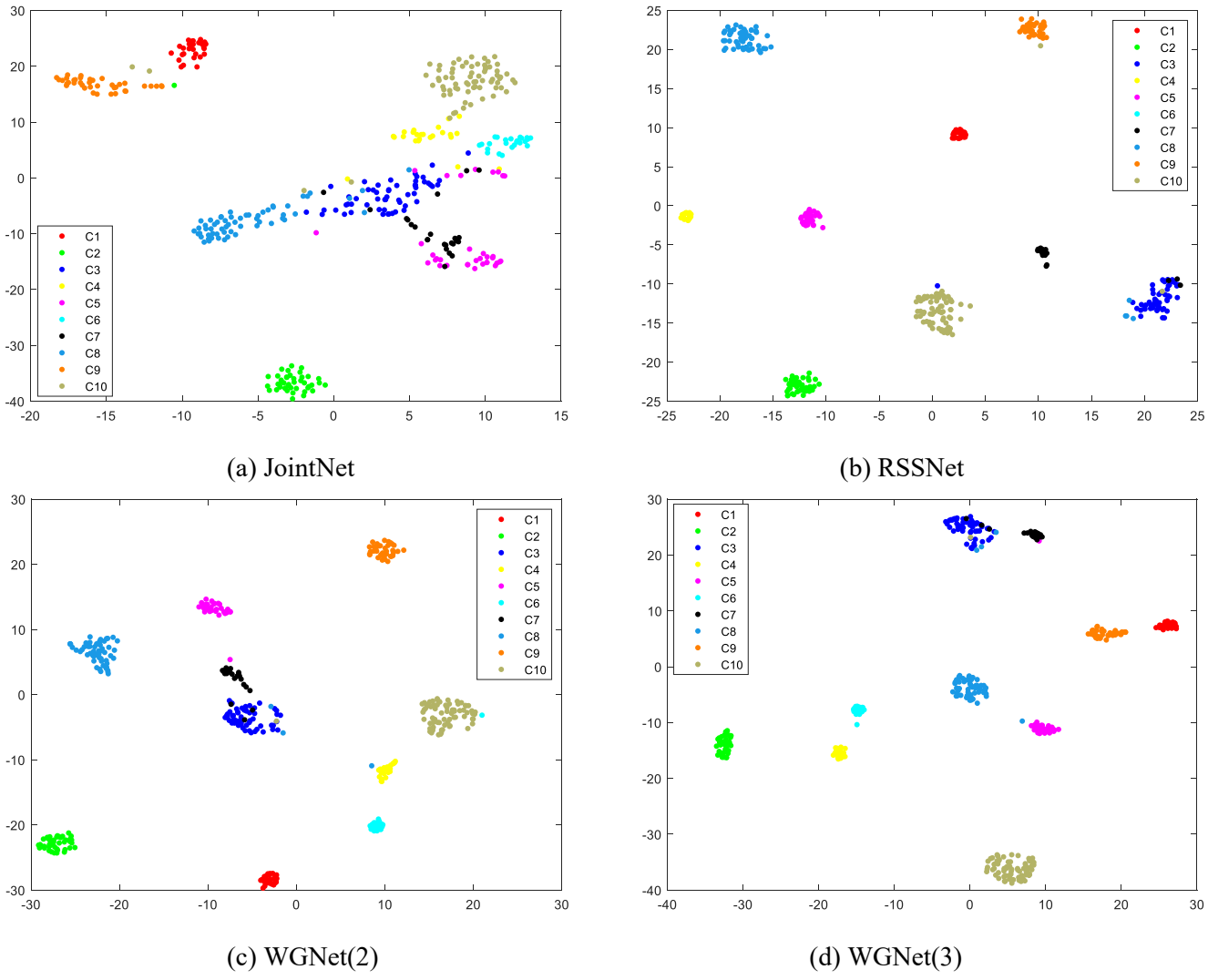


Fig. 7. Visualization of representation learned by different methods on subset 1, where C_i is class i in the dataset.

Table 6
The recognition performance comparison results of different methods on four subsets of ModelNet40.

Dataset	Metric	PointNet	DGCNN	WGNet(2)	WGNet(3)
Subset1	MA	88.91	92.24	91.24	**92.73**
	OA	89.62	92.42	92.17	**93.29**
Subset2	MA	80.37	81.69	83.89	**84.42**
	OA	85.55	87.48	87.67	**88.82**
Subset3	MA	88.46	88.13	**92.32**	91.28
	OA	91.92	91.11	**92.61**	92.20
Subset4	MA	84.60	83.30	90.90	**91.50**
	OA	91.14	90.94	95.57	**95.95**

3.4.3. Efficiency of CIA module

In CIA module, as one of the data preprocessing stages, the PCA related operations, including the eigen-decomposition, will affect the model's computational complexity. We conduct experiments to evaluate the influence. Specifically, since the number of points (i.e., n) in a point cloud and the number of neighbor points (i.e., N_{knn}) for each point are two main factors in PCA operation, we report the time consumption with different values of these two factors. The value ranges of n and N_{knn} are set to {1000, 3000, 5000, 7000, 9000, 10000} and {10, 20, 30, 40, 50}, respectively. Fig. 10 shows the results. Obviously, the results of PCA operation with different values of N_{knn} are similar, which means that the number of neighbor points has little effect on the efficiency of PCA operation. In addition, when the number of neighbors is fixed, the time consumption of PCA operation increases linearly with the increase of the number of points. This result show that the time complexity of PCA operation in CIA module can be considered as O(n), where n is the number of points, which is acceptable in practice.

3.5. Robustness analysis

We conducted experiments to evaluate the robustness of WGNet to common disturbances under road environments, including the Gaussian noise, down-sampling, and outliers. δ, η, and λ are denoted the standard deviation of Gaussian noise, down-sampling ratio, and outliers' rate, respectively. The ranges are {0.01, 0.02, 0.03, 0.04, 0.05}, {0.5, 0.25}, and {0.05, 0.10, 0.15, 0.20, 0.25}, respectively. PR curve is used to evaluate the performance. We investigate the robustness of WGNet(2).

1) Robustness to Gaussian noise.

As shown in Fig. 11(a), WGNet(2) can achieve excellent performance as $\delta \leq 0.02$. However, as δ continues to increase, the performance of WGNet(2) decrease sharply. The main reason is that with the increase of δ, the damage of Gaussian noise on the shape of objects increases, which leads to the change of point distribution pattern, so reduces the accuracy of recognition. In addition, for the influence of parameter K_{cheby}, it can be found that WGNet with $K_{cheby} = 3$ has better robustness. This is mainly

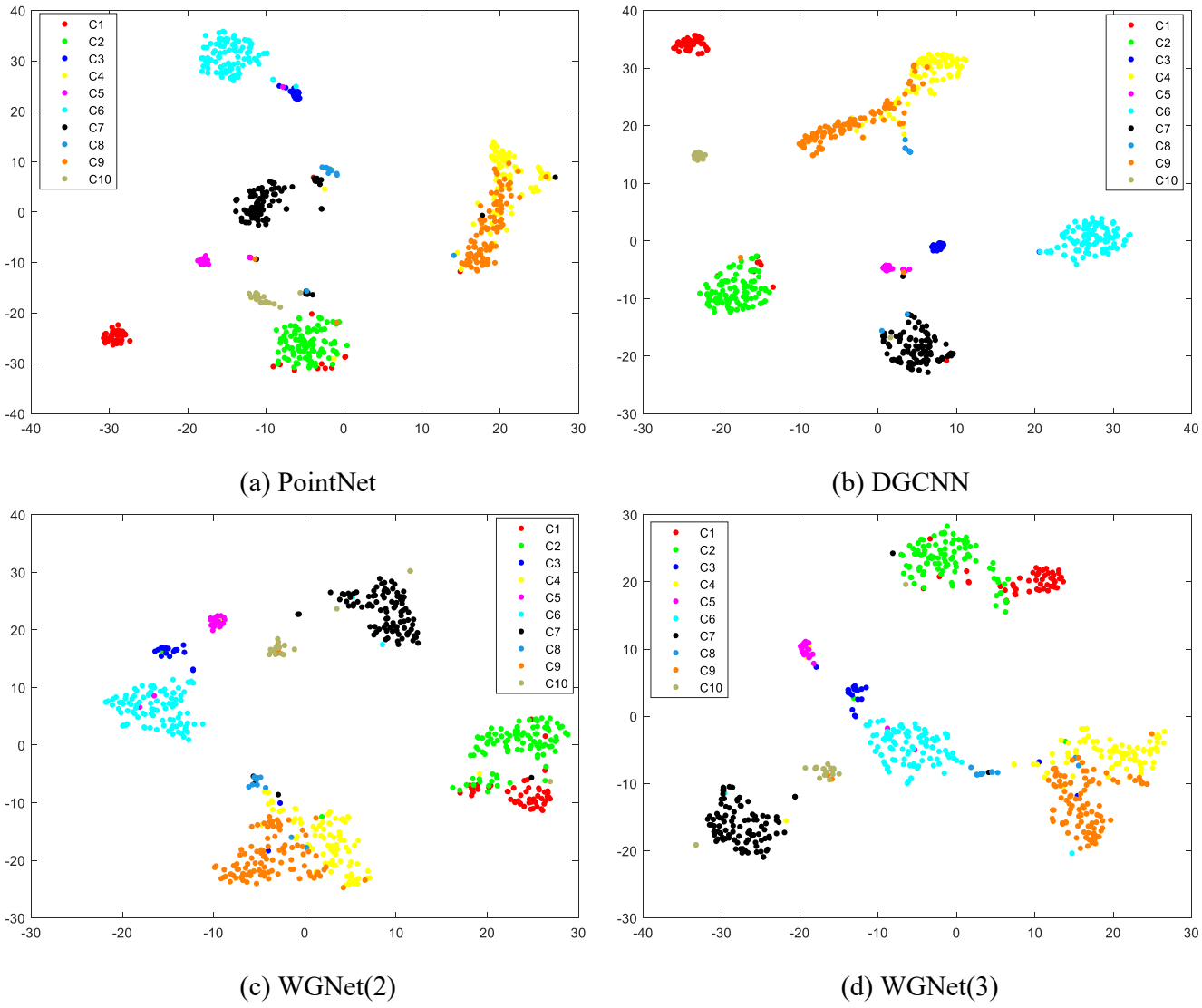


Fig. 8. Visualization of representation learned by different methods on subset 1, where C_i is class i in the dataset.

Table 7
Efficiency comparison of different methods on Sub-RObject10. G-M denotes the GPU-memory consumption. BW and FW stand for the backward, forward, respectively. k is K_{cheby}.

methods	Model size (MB)	G-M (MB)	BW (ms)	FW (ms)
PointNet	2.68	**919**	7.2	2.02
DGCNN	**6.93**	1513	45	20
WGNet(2), $k = 1$	0.59	1377	10.5	3.3
WGNet(2), $k = 2$	1.09	1395	11.7	3.4
WGNet(2), $k = 3$	1.60	1413	12.2	3.9
WGNet(3), $k = 1$	1.16	1415	11.7	4.7
WGNet(3), $k = 2$	2.22	1431	12.8	4.9
WGNet(3), $k = 3$	3.29	1449	13.2	5.4

because when K_{cheby} increases, the local receptive field of the model will become large, which improves the robustness.

2) Robustness to down sampling.

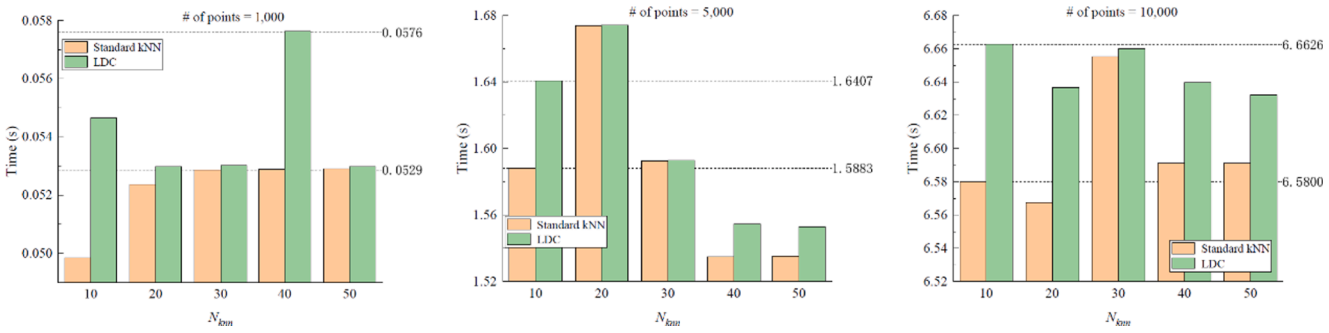
As shown in Fig. 11(b), WGNet can have a good performance with down sampling. When the density is down sampled to 64 points, the recall is still above 96% with the maximum accuracy. These results demonstrate the strong robustness of WGNet to down sampling. It can also be found that WGNet with $K_{cheby} = 3$ is weaker than that with $K_{cheby} = 2$. This is because point clouds become sparse with the increase of sampling rate, and expanding the receptive field cannot bring more useful information.

3) Robustness to outliers.

As shown in Fig. 11(c) and (d), the position of each PR curve decreases with the increase of outlier rate. This evidences that the increase of outlier rate impairs the performance of WGNet. In addition, it can be found that WGNet(2) obtains better results with $K_{cheby} = 3$ than that with $K_{cheby} = 2$. These results show that enlarging the receptive field is helpful to improve the robustness to outliers of the proposed model.

4) Improving robustness to outliers.

As discussed in Section 2.4.1, the maximum value is proposed as one of the local distribution characteristics. Considering that if the maximum value is an outlier, the result may be affected. Therefore, we conduct experiments to evaluate the effect of maximum value on the robustness of model with respect to outliers. Fig. 12 shows the compared results. Obviously, for WGNet(2), the model without maximum value has better robustness than the one with maximum value. Especially, with the increase of outliers disturbance, WGNet(2) without maximum value shows more significant superior performance. This result means that if the maximum value is an outlier, the robustness of WGNet(2) will be significantly affected. For WGNet(3), when the outliers rate is less than 0.2, the model with the maximum value achieves better results. However, when the outliers rate reaches 0.25, the accuracy of WGNet(3)



(a) # of points = 1000 (b) # of points = 5000 (c) # of points = 10000

Fig. 9. The compared results of different neighbor selection strategies with number of points = (a) 1,000, (b) 5,000 and (c) 10,000, respectively.

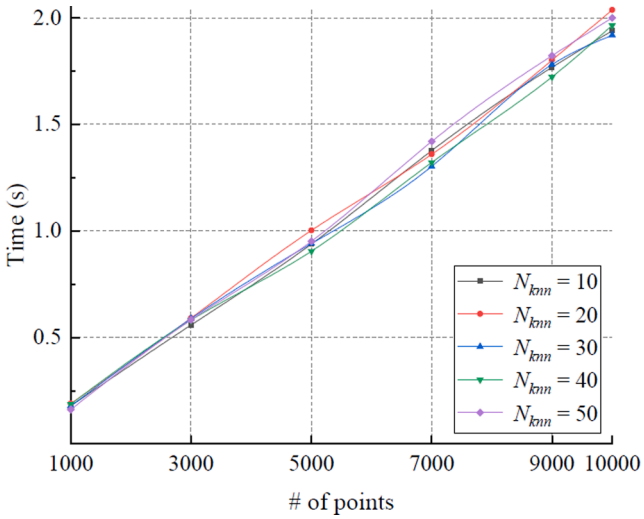


Fig. 10. Time consumption of PCA operation with different settings.

without the maximum is higher than that with the maximum. This shows that when the outliers disturbance is serious, the maximum value will affect the robustness of WGNet(3). These results demonstrate that if the data is disturbed by outliers, the maximum value can be removed in the local distribution characteristics.

3.6. Ablation study

3.6.1. Effects of LDC

In WGNet, the edge of graph is constructed using local dilated connecting, which is different from the common method, standard kNN search algorithm. We conducted comparison experiments for these two methods. The results are shown in Table 8. Obviously, LDC has significant advantage over the standard kNN. Especially, WGNet(2) has the most obvious improvement in terms of MA. Compared with standard kNN, WGNet(2) with LDC improves by 3.67%. To better analyze the results, PR curve is used for comparison. As shown in Fig. 13, PR curves of WGNet with LDC are of higher accuracy, while that of using standard KNN are lower. This means LDC has better performance. All these results show the feasibility and necessity of LDC.

3.6.2. Effects of CIA

In WGNet, the node signals include 3D coordinates and the neighborhood distribution features and local dimension features obtained by CIA module. To evaluate the effect of these additional node information, we conducted comparison experiments. As can be seen from Table 9, WGNet with CIA has significant advantage over that with only coordinate values. Especially, the improvement of WGNet(2) is about 10% and 9.57% in terms of MA and OA. PR curve is used to better compare the experimental results. As shown in Fig. 14, PR curves of WGNet with the node type of "coordinate value + context information" are higher than that of the coordinate values. These results demonstrate the effectiveness and necessity of CIA module.

It is also interesting to observe that WGNet with CIA has better performance than that with LDC. For example, as shown in Tables 8 and 9, WGNet(2) with CIA can obtain a gain of 9.89%, while WGNet(2) with LDC achieves improvement of 3.67% in terms of MA. The main reason is that, the information gain from expanding the receptive field through LDC is less than the gain from capturing more node information through CIA.

3.6.3. Effects of different edge connections and node representations

For comprehensive comparison of the influence of two designed modules, we conducted a comparative experiment without using two modules at the same time. Specifically, we tested the basic model that only uses kNN to construct edge connection and coordinate values as node features, then compared it with WGNet. Table 10 presents the results. Obviously, the proposed WGNet has better performance than the basic model. Especially, the OA and MA of WGNet(2) have been significantly improved with the gain of more than 10%. In addition, Fig. 15 shows the PR curves of the comparison results. It can be observed that the PR curves of WGNet has the higher position. These results further demonstrate the effectiveness of the designed two modules.

3.6.4. Applications with different backbones

1) As analyzed in the above section, in this work, the backbone can be replaced by other frameworks. To evaluate the possibility of extending our method, especially the CIA and LDC modules, for segmentation task, we take PointNet as backbone, and apply it on two public datasets, the ShapeNet (Wu et al., 2014) and S3DIS (Armeni et al., 2016), for shape part segmentation and scene semantic segmentation, respectively. IoU and OA are used to measure the performance. The compared results are shown in Table 11. We can observe that the two proposed modules can be applied for segmentation. Moreover, compared with the baseline (i.e., PointNet), our method achieves better performance on both two metrics. This is because the two proposed modules can expand the size of receptive fields and providing more information to describe samples accurately.

2) It is necessary and important to apply the proposed LDC and CIA modules to other backbones to validate the generalization. We replace the ChebyNet by DGCNN. Table 12 shows the compared results. Obviously, from Table 12 we can find that our method has significant advantages in terms of OA and MA. This is mainly because comparing with DGCNN, our method can not only enlarge the receptive field size via LDC module, but also provide more rich shape context information by CIA module. These results demonstrate the necessity and effectiveness of the LDC and CIA modules.

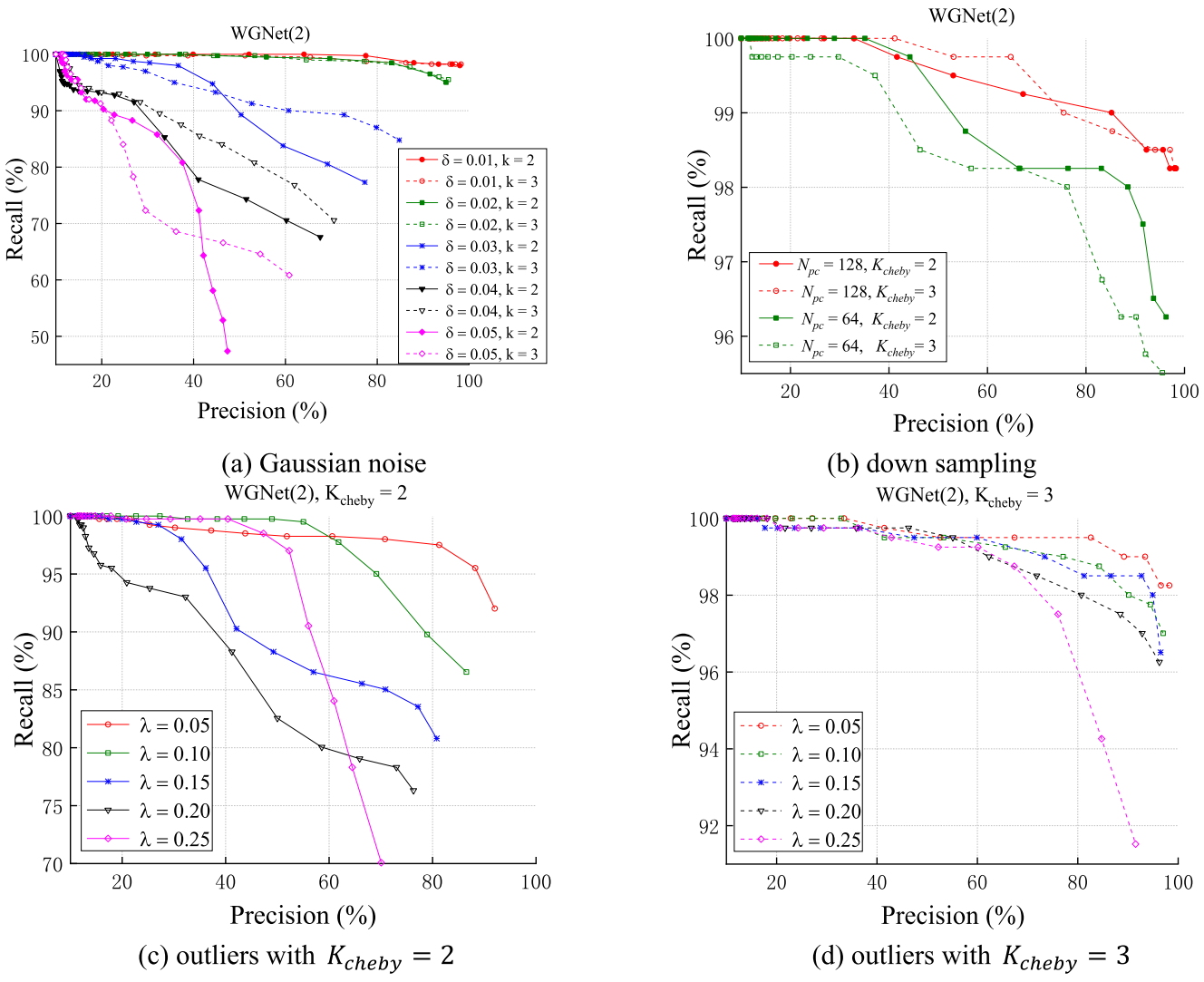


Fig. 11. PR curves with different disturbances: (a) Gaussian noise, (b) down sampling, (c) and (d) outliers, respectively.

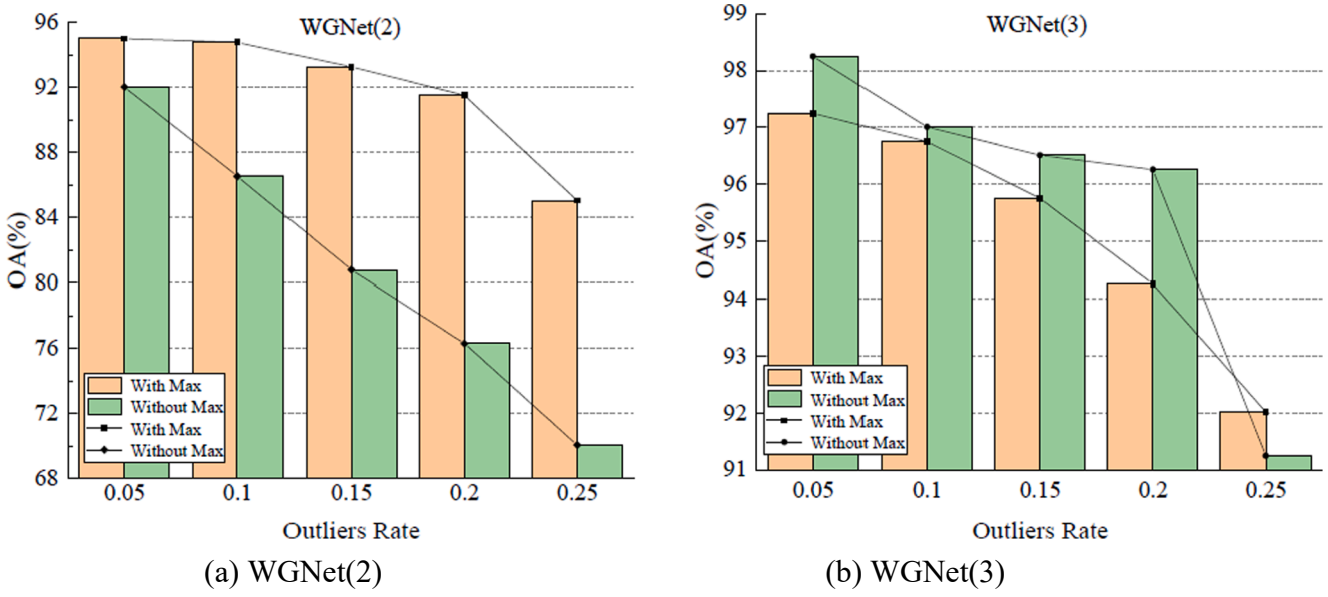


Fig. 12. Compared results of with/without maximum values in (a) WGNet(2), (b) WGNet(3), respectively.

Table 8
Comparison of different edge connections on Sub-RObject10.

Edge connection method	WGNet(2) MA (%)	OA (%)	WGNet(3) MA (%)	OA (%)
Standard method	94.25	94.51	95.97	96.26
Local dilated connecting	97.92 (↑3.67*)	98.00 (↑3.49)	98.10 (↑2.13)	98.25 (↑1.99)

*:The number in bracket is the margin of accuracy between two kinds of edge connection, where "↑" indicates increase.

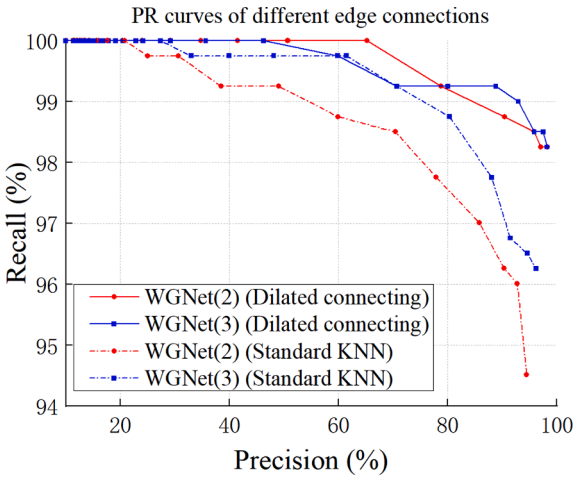


Fig. 13. PR curves with different edge connections.

Table 9
Comparison of different node representations on Sub-RObject10.

Node representation	WGNet(2) MA (%)	OA (%)	WGNet(3) MA (%)	OA (%)
Coordinate value	88.03	88.43	95.05	95.26
Coordinate value + context information	97.92 (↑9.89)	98.00 (↑9.57)	98.10 (↑3.05)	98.25 (↑2.99)

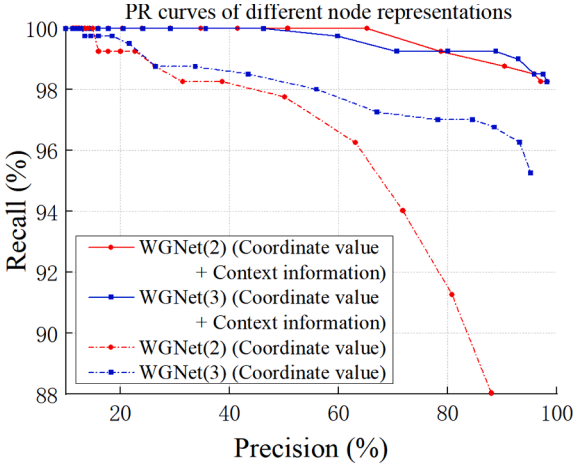


Fig. 14. PR curves with different node representations.

4. Discussion

Comparison results demonstrate the superior performance of the proposed method in terms of descriptiveness, efficiency and robustness. This can be summarized as for the following reasons: (1) The designed

Table 10
Comparison of different node representations on Sub-RObject10.

	WGNet(2) MA (%)	OA (%)	WGNet(3) MA (%)	OA (%)
Basic model	85.79	87.93	92.75	93.02
Our methods	97.92 (↑12.13)	98.00 (↑10.07)	98.10 (↑5.35)	98.25 (↑5.23)

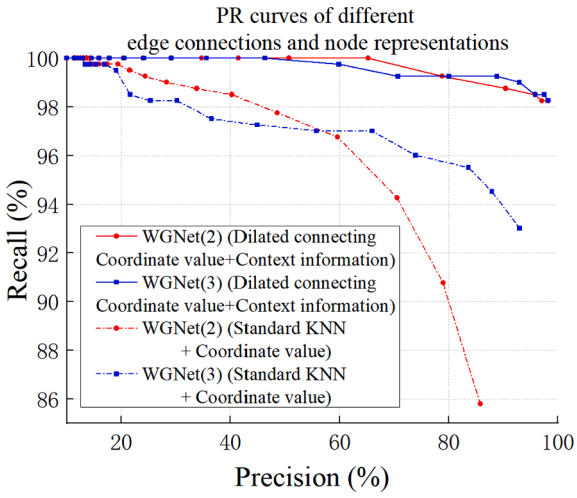


Fig. 15. PR curves with different edge connections and node representations.

Table 11
Results of part/scene segmentation generated by PointNet and the proposed method.

Method	Backbone	Part Segmentation IOU	OA	Scene Segmentation IOU	OA
PointNet	–	83.72	93.60	47.60	78.50
Our	PointNet	84.56	94.09	54.90	83.16

Table 12
Results of part/scene segmentation generated by PointNet and the proposed method.

Method	Backbone	# of points	MA	OA
DGCNN	–	1024	90.2	92.9
		2048	90.7	93.5
Ours	DGCNN	1024	91.54	93.53
		2048	92.31	93.79

LDC module constructs the adjacency matrix for the graph data. Compared with standard KNN search algorithm, LDC module can expand the receptive field of graph convolution to encode more information. (2) CIA module used in our method generates the node features as the initial input of GCNs. Different from normal initial input with just only three coordinate values, for each point, CIA module embeds the distribution patterns of its neighborhood points and local dimension features, resulting in stronger descriptiveness. (3) Build on ChebyNet, the proposed method has lightweight networks and few parameters. In addition, skip-connection is used to maximize the reuse of features learned by each level. Combined with LDC and CIA modules, an efficient, robust, and wider GCN, WGNet, is proposed to mine richer features to compensate the insufficiency on the depth of GCN.

Limitations. As discussed in section 3, in the training of WGNet, the Laplacian matrices of the graphs of training samples would be read into memory. Therefore, there are strict restrictions on the number of graphs

and nodes, which would make it difficult to apply WGNet in large-scale datasets. This will weaken its generalization. Additionally, as presented in section 3.5, although WGNet is robust to several kinds of disturbance, such as down sampling and outliers, it is still somewhat sensitive to Gaussian noise. In addition, if the point density is inhomogeneous, the results of KNN operation will be affected. More specifically, when the point density is homogeneous, with fixed k, the local points generated by KNN operation will describe the local shape exactly. However, if the point density is low/high, the region with fixed k points would be a large/small one, and the corresponding local point clouds may not mine the local point distribution pattern accurately. Consequently, the LDC and CIA modules may not capture helpful context information, so the performance of model would be affected. In practice, inhomogeneous point density remains in some regions, such as the occlusion and self-occlusion. Several ways can be used to reduce the effect of this issue. For example, the uniform down-sampling, such as FPS (Farthest Point Sampling), can be applied to obtain a homogeneous point density. Our future work will focus on resolving these shortcomings. Besides, since these model and parameters combinations used in this paper might rely on the distribution of dataset, we suggest users should re-test the performance of different parameter settings when applying the proposed method to other datasets, especially acquired by different scanners.

5. Conclusions

Aiming at the transformation from point clouds to graph structure, a graph building strategy based on local dilated connecting and context information awareness is proposed to construct the widen graph data. The feasibility and necessity of the proposed strategy are verified from the aspects of descriptiveness, efficiency, and robustness. In addition, the effects of LDC and CIA modules are analyzed by ablation experiments, which further verifies the effectiveness of the proposed method. Comparison results indicate the superiority of our architecture. In conclusion, we believe that our method provides a perspective of using the wider networks to compensate for the shortcoming of depth in GCNs.

CRediT authorship contribution statement

Yiping Chen: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Zhipeng Luo:** Conceptualization, Methodology, Writing – review & editing. **Wen Li:** Methodology. **Haojia Lin:** . **Abdul Nurunnabi:** . **Yaojin Lin:** Validation. **Cheng Wang:** Validation. **Xiao-Ping Zhang:** Validation. **Jonathan Li:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by National Natural Science Foundation of China (grants: 41871380), the Natural Sciences and Engineering Research Council of Canada (grant: 50503-10284). Dr Nurunnabi is funded as part of Project 2019-05-030-24, SOLSTICE - Programme Fonds Européen de Developpment Régional (FEDER)/Ministère de l'Economie of the G. D. of Luxembourg..

References

Armeni, I., Sener, O., Zamir, A., jiang, H., Brilakis, I., Fischer, M., Savarese, S. 2016. 3D semantic parsing of large-scale indoor spaces," In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 534–1543.

Chen, Y., Wang, J., Li, J., Lu, C., Luo, Z., Xue, H., Wang, C., 2018. LiDAR-Video Driving Dataset: Learning Driving Policies Effectively. In: In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, pp. 5870–5878.

Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y., 2020. Simple and deep graph convolutional network. In International Conference on Machine Learning. 1725–1735.

Feng, H., Li, W., Luo, Z., Chen, Y., Fatholahi, S.N., Cheng, M., Wang, C., Junior, J.M., Li, J., 2021. GCN-based pavement crack detection using mobile LiDAR point clouds. IEEE Trans. Intell. Transp. Syst. https://doi.org/10.1109/TITS.2021.3099023.

Seo, Y.W., Lee, J., Zhang, W., Wettergreen, D., 2015. Recognition of highway workzones for reliable autonomous driving. IEEE Trans. Intell. Transp. Syst. 16 (2), 708–718.

Gong, Z., Li, J., Luo, Z., 2019. Mapping and semantic modeling of underground parking lots using a backpack LiDAR system. IEEE Trans. Intell. Transp. Syst. 99, 1–13.

Chen, P., Shi, W., Fan, W., Xiang, H., Bao, S., 2021. RectMatch: a novel scan matching method using the rectangle-flattening representation for mobile LiDAR systems. ISPRS J. Photogramm. Remote Sens. 180, 191–208.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point clouds. In: In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 11105–11114.

Thomas, H., Qi, C., Deschaud, J., Marcotegui, B., Goulette, F., Guibas, L., 2019. KPConv: Flexible and deformable convolution for point clouds. In: In: IEEE International Conference on Computer Vision. IEEE, pp. 6410–6419.

Yang, Z., Sun, Y., Liu, S., Jia, J., 2020. 3DSSD: Point-Based 3D Single Stage Object Detector. In: In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 11037–11045.

Qi, C., Chen, X., Litany, O., Guibas, L., 2020. ImVoteNet: Boosting 3D Object Detection in Point Clouds With Image Votes. In: In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 4403–4412.

Xie, Q., Lai, Y., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J., 2020. MLCVNet: Multi-Level Context VoteNet for 3D Object Detection, In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 10444-10453.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449.

Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: European Conference on Computer Vision. Springer, Prague, Czech Republic, pp. 224–237.

Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. Int. J. Comput. Vision 105 (1), 63–86.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2014. 3D ShapeNets: a deep representation for volumetric shapes. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Columbus, Ohio, pp. 1912–1920.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. In: IEEE International Conference on Intelligent Robots and Systems. IEEE, Hamburg, Germany, pp. 922–928.

Su, H., Maji, S., Kalogerakis, E., Learnedmiller, E., 2015. Multi-view convolutional neural networks for 3D shape recognition. In: IEEE International Conference on Computer Vision. IEEE, Santiago, Chile, pp. 945–953.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Las Vegas, Nevada, pp. 77–85.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic graph CNN for learning on point clouds. ACM Trans. Graphics 38 (5), 1–12.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L.i., Bennamoun, M., 2020. Deep learning for 3D point clouds: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. NeurIPS. 3844–3852.

Te, G., Hu, W., Zheng, A., Guo, Z., 2018. RGCNN: Regularized graph CNN for point cloud segmentation. ACM Multimedia. 746–754.

Li, R., Wang, S., Zhu, F., Huang, J., 2018. Adaptive graph convolutional neural networks. Conf. Artif. Intell. AAAI 3546–3553.

Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral networks and locally connected networks on graphs. Int. Conf. Learning Represent. ArXiv:1312.6203v3.

Yi, L., Su, H., Guo, X., L. Guibas, J., 2017. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2282–2290.

Wang, C., Samari, B., Siddiqi, K., 2018. Local spectral graph convolution for point set feature learning. Eur. Conf. Comput. Vision Workshops 52–66.

Zhang, Y., Rabbat, 2018. A Graph-CNN for 3D Point Cloud Classification. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp. 6279-6283.

Pan, G. h., Liu, P., Wang, J., Ying, R., Wen, F., 2019. 3DTI-Net: Learn 3D Transform-Invariant Feature Using Hierarchical Graph CNN. Pacific Rim International Conference on Artificial Intelligence. pp. 37–51.

Simonovsky, M., Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 29–38.

Zhang, K., Hao, M., Wang, J., Silva, C. W., Fu, C., 2019a. Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features. ArXiv:1904.10014.

Zhang, Z., Peng, H., 2019b. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In: In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 4586–4595.

Liu, J., Ni, B., Li, C., Yang, J., Tian, Q., 2019. Dynamic points agglomeration for hierarchical point sets learning. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 7546–7555.

Shen, Y., Feng, C., Yang, Y., Tian, D., 2018. Mining point cloud local structures by kernel correlation and graph pooling. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 4548–4557.
Dominguez, M., Dhamdhere, R., Petkar, A., Jain, S., Sah, S., Ptucha, R., 2018. General-purpose deep point cloud feature extractor. In: Winter Conference on Applications of Computer Vision. IEEE, pp. 1972–1981.
Komarichev, A., Zhong, Z., Hua, J., 2019. A-CNN: Annularly convolutional neural networks on point clouds[C]. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 7421–7430.
Rong, Y., Huang, W., Xu, T., Huang, J., 2019. Dropedge: Towards deep graph convolutional networks on node classification. In: ICLR.
Lu, Z., Pu, H., Wang, F., Hu, Z., Wang, L., 2017. The Expressive Power of Neural Networks: A View from the Width. In: In: Conference and Workshop on Neural Information Processing Systems. ACM, pp. 9565–16240.
Shang, W., Sohn, K., Almeida, D., Lee, H., Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units, 2016. In: International Conference on Machine Learning. ACM, pp. 2217-2225.
Huang, G., Liu, Z., Maaten, L., Weinberger, K., Densely Connected Convolutional Networks, 2017. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2261-2269.
Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions. Int. Conf. Learning Represent. ArXiv:1511.07122v3.
Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. ArXiv:1511.07122v3, 1–14.
Gressin, A., Mallet, C., Demantke, J., David, N., 2013. Towards 3D LiDAR point cloud registration improvement using optimal neighborhood knowledge. ISPRS J. Photogramm. Remote Sens. 79, 240–251.
Lin, H., Chen, J., Su, P., Chen, C., 2014. Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification. ISPRS J. Photogramm. Remote Sens. 94, 70–79.
Luo, Z., Liu, D., Li, J., Chen, Y., Xiao, Z., Junior, J.M., Gonçalves, W.N., Wang, C., 2020. Learning sequential slice representation with an attention-embedding network for 3D shape recognition. ISPRS J. Photogramm. Remote Sens. 161, 147–163.
Luo, Z., Li, J., Xiao, Z., Mou, G., Cai, X., Wang, C., 2019. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. ISPRS J. Photogramm. Remote Sens. 150, 44–58.