



This is a repository copy of *Long short-distance topology modelling of 3D point cloud segmentation with a graph convolution neural network*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/194549/>

Version: Published Version

Article:

Zhang, W.J., Su, S.Z., Hong, Q.Q. et al. (2 more authors) (2022) Long short-distance topology modelling of 3D point cloud segmentation with a graph convolution neural network. IET Computer Vision. ISSN 1751-9632

<https://doi.org/10.1049/cvi2.12160>

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown


If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

ORIGINAL RESEARCH

Long short-distance topology modelling of 3D point cloud segmentation with a graph convolution neural network

 Wen Jing Zhang¹ | Song Zhi Su¹  | Qing Qi Hong¹ | Bei Zhan Wang¹ | Li Sun²
¹School of Informatics, Xiamen University, Xiamen, China

²Department of Computer Science, The University of Sheffield, Sheffield, UK

Correspondence

Song Zhi Su and Qing Qi Hong, School of Informatics, Xiamen University, Xiamen 361005, China.

 Email: ssz@xmu.edu.cn and hongqq@xmu.cn

Abstract

3D point cloud segmentation is a non-trivial problem due to its irregular, sparse, and unordered data structure. Existing methods only consider structural relationships of a 3D point and its spatial neighbours. However, the inner-point interactions and long-distance context of a 3D point cloud have been less investigated. In this study, we propose an effective plug-and-play module called the Long Short-Distance Topologically Modelled (LSDTM) Graph Convolutional Neural Network (GCNN) to learn the underlying structure of 3D point clouds. Specifically, we introduce the concept of subgraph to model the contextual-point relationships within a short distance. Then the proposed topology can be reconstructed by recursive aggregation of subgraphs, and importantly, to propagate the contextual scope to a long range. The proposed LSDTM can parse the point cloud data with maximisation of preserving the geometric structure and contextual structure, and the topological graph can be trained end-to-end through a seamlessly integrated GCNN. We provide a case study of triple-layer ternary topology and experimental results on ShapeNetPart, Stanford 3D Indoor Semantics and ScanNet datasets, indicating a significant improvement on the task of 3D point cloud segmentation and validating the effectiveness of our research.

1 | INTRODUCTION

Different from 2D image, 3D data has several popular representations, such as polygonal mesh, registration of multiple images, and point cloud. The most common are polygonal mesh and point cloud. Polygonal mesh exploits connectivity information to describe 3D data. Point cloud consists of a set of points, each of which is composed of 3D coordinates and possibly attributes, for example, colour and normal. Point cloud is a generic and most widely used representation of 3D data that has drawn increasing popularity in a broad range of applications, for example, robotic mapping, autonomous vehicle, and navigation [1–3]. With the popularity of the range sensors, for example, Kinect, Lidar, radar, semantic understanding of point cloud is a foundational application for robotics and automotive [4–6]. Unlike 2D image, 3D point cloud is a set of unstructured and unordered points of non-unified numbers, which makes the existing 2D methods less effective in representation and learning.

Inspired by the success of Convolutional Neural Network (CNN) on 2D images, the point cloud is transformed into voxels or multi-view data for the adaptation of 3D CNN technology. However, 3D CNNs [7, 8] and 2D multi-view CNNs [9–11] have experienced a significant consumption in term of computation and memory efficiency due to the sparsity of point cloud. Recently, various encouraging methods directly parsing the point clouds from 3D raw points have been proposed, such as PointNet [12]. To effectively learn semantically relevant information, researchers have proposed a series of methods based on the PointNet algorithm. These methods can be generally categorised as: (1) neighbouring feature learning [13–22]; (2) optimising CNN [23–27]; (3) recurrent neural network (RNN) [28, 29]; (4) attention-based aggregation [30–33]; (5) combined instance segmentation [34–38]; (6) graph convolution [39–48]. Although these methods have achieved impressive performances for object segmentation and semantic segmentation, almost all of them are limited to interpret implicit contextual-point interaction and long-distance contextual

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *IET Computer Vision* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

relationship using a generic representation. Exploring the long-distance dependencies relationships of points in 3D point cloud is indeed important for understanding 3D scenes [48, 49]. In order to better investigate the complex interactions and long-distance contextual relationships of points in 3D point cloud, we propose an effective plug-and-play module called the Long Short-Distance Topologically Modelled (LSDTM) Graph Convolutional Neural Network (GCNN) for 3D point cloud segmentation, named LSDTM-GCNN. Inspired by the recent DS-CAE network [50], LSDTM-GCNN also decomposes a graph into a family of k -layer expansion graphs rooted at each vertex, aiming at better capturing the inter-dependencies of long-range vertices. In this paper, we formulate the point cloud contextual information discovery as a task of modelling inter-dependencies of long-range vertices in a topological graph. With the consideration of the fact that points in a point cloud are of a very large number in real-world applications, learning the long-range contextual relationship using a fully connected graph is NP-hard and unlikely to be computationally tractable.

To mitigate this issue, we propose a generic multi-layer multiple-arg topological architecture to represent the down-sampled points. Our approach focuses on discovering the underlying knowledge to connect the low-layer geometrical features with high-layer inference. Note, it can be built above the network architectures that focus on the local feature learning, for example, PointNet++. Main contributions of our paper are:

- 1) We propose a generic topological graph for point cloud representation, that is, LSDTM, to interpret the implicit long-distance contextual-point relationship by discovering the inner-dependencies of long-distance vertices in a graph.
- 2) Our proposed LSDTM enables GCNN to effectively propagate through the dedicated designed architecture to optimise the topological graph and discover the underlying relationships. The learnt LSDTM representation is geometric and contextual enriched with preservation of spatial contiguity, which can significantly facilitate the point cloud segmentation and understanding.
- 3) Our approach achieves comparable results with fair runtime performance in widely cited benchmarks for both object part segmentation and semantic segmentation.

2 | RELATED WORK

Recently, researchers have proposed a number of methods to handle semantic segmentation of 3D point clouds. Depending on the different ways of dealing with point clouds, we generally classify these methods into two categories: deep learning on ordered sets and deep learning on raw points sets.

2.1 | Deep learning on ordered sets

Inspired by 2D CNNs, many methods [51–54] apply voxelisation to point cloud segmentation tasks. The volumetric representation naturally preserves the neighbour structure of

the 3D point cloud. In addition, to further exploit the powerful representation capabilities of 2D CNNs, [10, 11] capture 2D images from multiple perspectives and then use a multi-view representation and some voting or fusion mechanism to complete the point cloud segmentation task. Although these methods have achieved good performance in point cloud semantic segmentation tasks, they are difficult to be applied to large-scale 3D point cloud scenes due to the limitation of computational efficiency.

2.2 | Deep learning on raw points sets

Charles et al. [12] propose a simple but effective deep learning model for point clouds, PointNet, and verify that it can be used for various cognitive tasks with point clouds. However, this method does not effectively take into account the local structural features between points in the point cloud. Following PointNet, many methods [13–16, 19, 20] have been proposed to explore geometric structure. Qi et al. [13] design a hierarchical neural network to better explore local structures. Similarly, Li et al. [14] use hierarchical feature extraction to explicitly model the spatial distribution of point clouds. Zhao et al. [15] design a new feature adaptive adjustment (AFA) method for extracting contextual features from local neighbourhoods. Inspired by 2D SIFT, Jiang et al. [16] design a PointSIFT module to encode information from eight different orientations of each point. Hu et al. [19] use random sampling to process large scale point clouds, resulting in significant improvements in both memory and algorithmic efficiency. However, random sampling is prone to discard critical information, especially for objects with sparse points. Unlike the above approaches, Fan et al. [20] construct a spatial representation that is invariant to Z -axis rotation to facilitate the learning of spatial features from large-scale point clouds. Overall, these methods are similar in that they rely on neighbours to obtain local features of the point cloud. Although they all achieve excellent performance, most of them are limited in revealing the underlying structure of 3D point clouds.

Li et al. [23] explore the idea of using equivariance function for 3D point clouds. However, it is limited in achieving the structural relations between local sub-clouds, which is desired for point clouds. Subsequently, Yifan et al. [25] design a special family of convolutional filters that apply different weights to each neighbour to extract semantic deep features. Komarichev et al. [26] design annular convolution to better capture the local neighbourhood geometry information of each point in the 3D point cloud. Refs. [17, 30, 41, 55–57] define a set of sharing weights applied to the neighbours of each vertex. Ref. [55] is a pioneering work of graph convolution in point cloud segmentation, which uses Multi-Layer Perceptrons (MLPs) to dynamically learn the convolution filters between connected vertices. Wang et al. [41] extend pointnet by proposing an edge convolution operation (EdgeConv) that captures the local geometric structure of the point cloud while maintaining permutation invariance. Subsequently, Wang et al. [17] extend pointnet++ by enrich each point representation, which can better explore the enriching neighbourhood of points and global information.

Inspired by graph attention networks, Chen et al. [30] design GAPNet to capture contextual attention features by distinguishing the importance of different neighbours to each point. Xun Xu [56] use a spatial and colour smoothing to encourage the prediction of points with similar colours. Xu et al. [57] design a fast sampling method, Coverage Ware Grid Query (CAGQ), which improves spatial coverage well and then uses Grid Context Aggregation (GCN) for information fusion. Most of these methods only construct local graphs based on points and their neighbours, and do not contribute towards discovering the underlying relationships and interactions of contextual points in 3D space. Our approach uses similar techniques to define convolutions over graph-structured data, in particular, our method uses LSDTM to explore the relationships between the points, and obtain enriched layer feature to preserve spatial contiguity of 3D point clouds.

3 | METHOD

3.1 | Method overview

Given a 3D point cloud $P = \{p_0, \dots, p_i, \dots, p_N\}$ with $p_i \in \mathbb{R}^d$ and a candidate label set $L = \{l_0, l_1, \dots, l_m\}$. Each point p_i contains (p_{ix}, p_{iy}, p_{iz}) coordinates and additional feature channels such as colour vectors, normal etc. Although additional feature channels can significantly improve classification performance, we only use the (p_{ix}, p_{iy}, p_{iz}) coordinate as inputs of our approach to demonstrate the ability of geometric structure learning. Firstly, we sample a subset $\hat{P}_t = \{p_j | j = 0, 1, \dots, t-1\}$ for the point cloud P . \hat{P}_t contains t points to define the local regions of centroid by iterative Fastest Point Sampling (FPS), here $2 < t \leq N + 1$. Secondly, the Ball Query [13] is used to search for all points that are within a radius R for each centroid. Next, MLP and Global Pooling Layer are used to obtain the enriched local features of t centroids respectively. These local feature vectors encode most of the local information and can be expressed as $x_t = \{\phi(p_j) | j = 0, 1, \dots, t-1\}$. We then propose the LSDTM-GCNN topological architecture to learn the deterministic long-distance geometrical and spatial relations of these centre points. The input to LSDTM-GCNN is the original coordinates and the feature embedding of the centre point. After iteratively applying the processing of FPS, Ball Query and LSDTM-GCNN Modules for multiple times, the centre points are aggregated to a smaller number but the features are enriched. Through multi-layer graph propagation, the obtained features have both representative geometrical shapes and spatial relationships. Then, the original point set is obtained using Feature Propagation [13] method. Finally, a fully connected layer is used to segment the frames. Details are shown in the Figure 1.

In Figure 1, the top panel gives the illustration of LSDTM-GCNN module. In this module, we show the construction process of the n -layer m -ary subgraph for each centre point, and the channel convolution operation of each subgraph. The LSDTM-GCNN module can better capture the interdependencies of long vertices. The details of this module are

illustrated in Sections 4 and 5. The bottom panel of Figure 1 depicts the overall architecture of our approach for point cloud part segmentation and semantic segmentation.

3.2 | Long short-distance topological modelling

We now explain the details of our approach by a generic graph representation of 3D point clouds that interprets the implicit long-distance contextual-point relationship. When the t centroids and their local neighbours are obtained, we use them to learn the enriched local features $x_t = \{\phi(p_j) | j = 0, 1, \dots, t-1\}$ by MLP and Global pooling. Then, the t centroids and x_t are used to construct the LSDTM representation of the subgraph $G_i = (V_i, E_i)$ composed of the centre point and its neighbours. We believe that the LSDTM representation of G_i can help the network learn the long short-distance interactions between the contextual points in 3D point clouds. Inspired by ref. [50], we develop a (n -layer m -ary) LSDTM representation of G_i for each centre point. The vertices in G_i are represented by the local feature vectors of the centre points. Unlike ref. [50], firstly, the vertices in our subgraph contain enriched local features. Secondly, our subgraph considers the edge information of the vertices, thereby the topological information of points in point cloud can be better learnt.

Specifically, given a centre point $p_i \in \hat{P}_t$, the construction of subgraph $G_i(V_i, E_i)$ for a vertex p_i consists of two steps:

- 1) finding m neighbours in R^d as m leaf vertices for the root vertex p_i using knn. The m neighbours of p_i is defined by p_{im} . It can be indicated as:

$$p_{im} = \{p'_{im} \mid \text{dist}(p_i, p'_{im}) < d\} \quad (1)$$

where $p'_{im} \in \hat{P}_t$, $m < t$ and $d \in R$ is the chosen radius.

- 2) each leaf vertex of p_{im} becomes the new root vertex, and then we further find their own m leaf vertices as the root vertices of the next layer of the subgraph $G_i(V_i, E_i)$. Here, the m neighbours of the h th leaf vertex of p_{im} is defined as p_{ihm} , which can be formulated as:

$$p_{ihm} = \{p'_{ihm} \mid \text{dist}(p_{im}, p'_{ihm}) < d\} \quad (2)$$

where $h = \{1, 2, \dots, m\}$. Similarly, the m neighbours of the h th leaf vertices for the root vertex p_{ihm} can be expressed as:

$$p_{ihbm} = \{p'_{ihbm} \mid \text{dist}(p_{ihm}, p'_{ihbm}) < d\} \quad (3)$$

Via repeating this operation, the (n -layer m -ary) subgraph G_i can be recursively constructed for p_i . Then, the corresponding local feature vectors of the vertices are selected in G_i as the vertices V_i of subgraph. In this way, the (n -layer m -ary) LSDTM representation of G_i is constructed. The vertices V_i of G_i can be formulated as:

$$V_i = \{\phi(p_i), \phi(p_{ihm}), \phi(p_{ihbm}), \dots, \phi(p_{ib\dots bm})\} \quad (4)$$

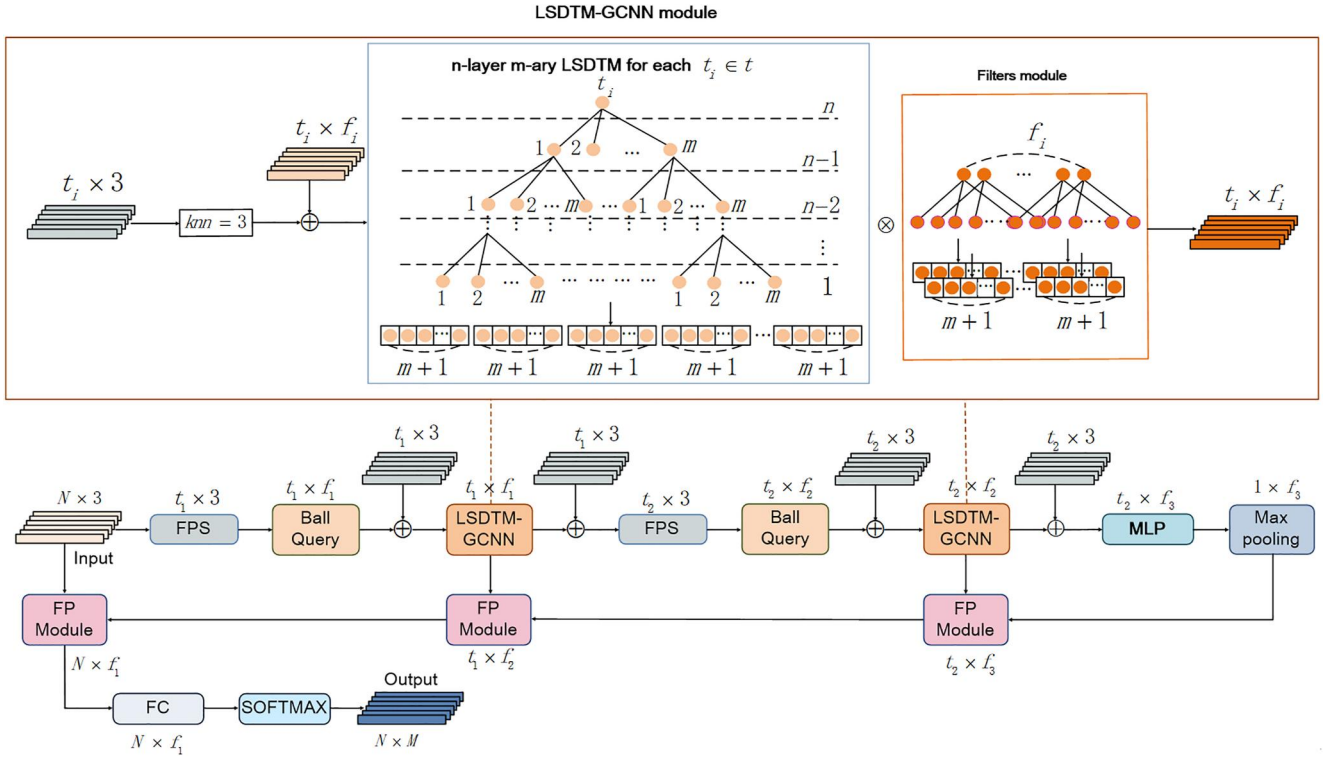


FIGURE 1 The top panel shows the LSDTM-GCNN module. In this module, we show a generic (n -layer m -ary) subgraph for each centre point p_i , $t_i \in t$ (Note a triple-layer trinary graph is used in our approach). Here, $t \in N$. Then, for each point p_i , its (n -layer m -ary) subgraph is convolved with the number of f_i filters with m -ary layer by layer. Finally, the long-distance contextual information of the points can be captured. The bottom panel shows the architecture of our method for point cloud part segmentation and semantic segmentation. Firstly, the enriched local features are obtained by FPS and Ball Query. Then, we use LSDTM-GCNN to capture the long-distance contextual relationship of the points. After several steps of processing by FPS, Ball Query and LSDTM-GCNN modules alternately, point feature are aggregated by max-pooling. Then, we use FP method to obtain the original set of points. Finally, FC and softmax layers are used to obtain the output. FC, fully connected; FP, feature propagation; FPS, fastest point sampling; GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled; MLP, multi-layer perceptron

The directed edge E_i consist of the root vertices to their leaf vertices. For example, $E_i(\phi(p_i), \phi(p_{im}))$ is defined as:

$$E_i(\phi(p_i), \phi(p_{im})) = H(\phi(p_i), \phi(p_i) - \phi(p_{im})) \quad (5)$$

The crucial part in Equation (5) is the definition of directed edge function H , which concatenates both the global shape structure (captured by the local feature of the centre points p_i) and local structure information (captured by $\phi(p_i) - \phi(p_{im})$). In each level of G_i , the connection between the roots and their leaf vertices maintain the initial short-distance spatial structure of the point cloud. The key to our LSDTM-GCNN is to define a GCNN to propagate the connections of the short-distance spatial structure to the long-distance. The details will be described in the Section 5. Algorithm 1 shows pseudo-code for the LSDTM representation of the subgraph for the point in 3D point cloud.

Algorithm 1 LSDTM representation for the point in 3D point cloud

Input: A point cloud P with $N + 1$ points, (n -layer m -ary)

Output: G_t subgraphs with (n -layer m -ary) LSDTM

```

1: initialisation;
2: Downsample  $\hat{P}_t = \{p_j | j = 0, \dots, t-1\}$  from  $P$ 
   using FPS;
3: for  $i = 0; i \leq t-1; i++$  do
4:   Compute the enriched local feature
   vector  $x_i = \phi(\hat{P}_i)$ ;
5:   Construct an  $m$ -ary tree of  $\hat{P}_i$  according
   to knn;
6:   Select  $x_i$  of the corresponding
   vertices in  $m$ -ary tree as the vertices;
7:   Compute the edge features for root
   vertex  $x_i$  and its  $m$ -ary vertices;
8:   for  $h = n-1, h >= 2$  do
9:     The leaf vertices of the ( $n$ -layer
      $m$ -ary) LSDTM are further replaced
     by their own  $m$ -ary vertices;
10:    Compute the edge feature of leaf
    vertices and their own  $m$ -ary
    vertices;
11:   end for
12: end for
13: return  $G_t$  subgraphs and each subgraph
   with ( $n$ -layer  $m$ -ary) LSDTM;

```

3.3 | LSDTM-GCNN

Figure 2 shows an illustration of our subgraph convolutional network for the *triple-layer ternary* LSDTM. Taking the centre point p_i as an example, Figure 2a gives the triple-layer ternary G_i for p_i . Here, $p_{i,3}$ is 3th layer vertex of the *triple-layer ternary* for G_i . $\{p_{i,2,j}|j = 1, 2, 3\}$ represents 2th layer j th vertices of the *triple-layer ternary* for G_i . Similarly, $\{p_{i,1,j}|j = 1, 2, 3, \dots, 9\}$ is 1th layer j th vertices of the *triple-layer ternary* for G_i . After obtaining G_i of p_i , we select the feature vector x of the corresponding vertex in the (3-layer 3-ary) subgraph as the vertex of G_i . Next, we compute the direct feature edges of G_i . Thus, the (3-layer 3-ary) LSDTM for G_i can be obtained as shown in Figure 2b left. Here, $x_{i,j,m}$ is the m th leaf vertex of j th-layer for the i th input root vertex and $E_{i,j,m,i,(j-1),m}$ represent the direct features edges of the m th leaf vertex of the j th-layer for x_i to its m th leaf vertex of the $(j-1)$ th-layer for x_i . Subsequently, the GCNN is applied layer by layer on the (3-layer 3-ary) LSDTM G_i for p_i to extract the hidden feature information. As shown in Figure 2b,c, convolution filter f_{n1} slides over each root vertex and its leaf vertices of subgraph G_i to obtain the hidden feature information of each root vertex, as shown in Figure 2d left. Where, $X_{i,3}$ is the hidden feature information of the root vertex $x_{i,3}$ and $X_{i,2,j}$ is the hidden feature information of the root vertex $x_{i,2,j}$. Then, the convolution filter f_{n2} is used to slide over the root vertex and its leaf vertices composed by hidden features, as shown in Figure 2d,e. Finally, we can obtain the hidden feature $\tilde{X}_{i,3}$, as shown in Figure 2f.

$$X_{i,(n-1),m}^f = \xi \left[\sum_{k=1}^{f_{n-2}} \left(\sum_{s=1}^{m+1} W_{k,s}^{n-1,f} \odot Q_{i,(n-1),m,s,k}^{n-2} \right) + B^{n-1,f} \right] \quad (6)$$

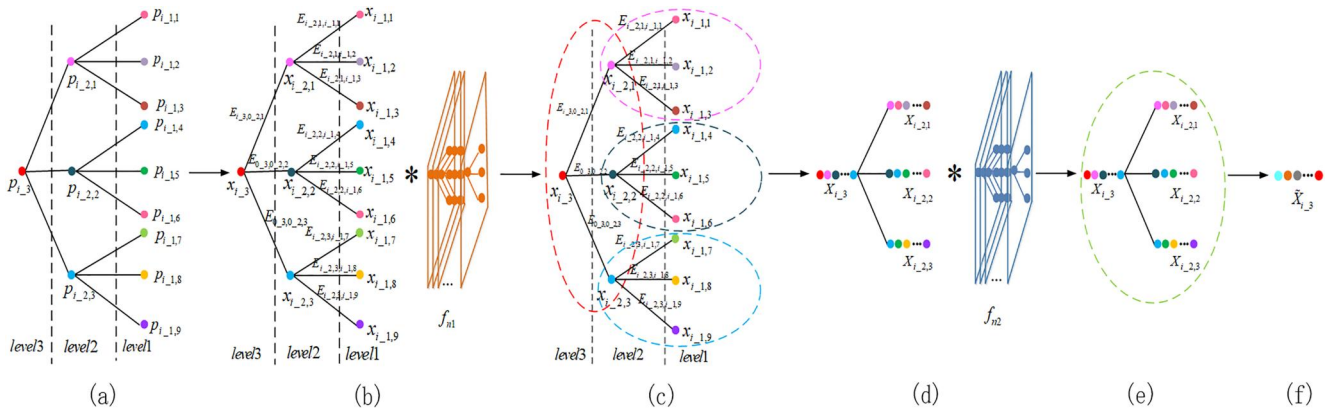


FIGURE 2 The framework of the subgraph convolution process for triple-layer ternary (3-layer 3-ary) long short-distance topologically modelled (LSDTM) is shown in detail. Here, we present an example of (3-layer 3-ary) subgraph G_i for centre point p_i , and construct the (3-layer 3-ary) LSDTM for subgraph G_i . Then, we design the number of f_{n1} and f_{n2} filters with 3-ary. From left to right, firstly, we construct (3-layer 3-ary) subgraph for centre point p_i as shown in (a). Secondly, we select the descriptive feature vector x of corresponding vertices in (3-layer 3-ary) subgraph as the vertices for G_i , and compute the direct features edges for G_i . The (3-layer 3-ary) LSDTM for subgraph G_i is shown in (b) left. Subsequently, (b), (c), (d), and (e) show the (3-layer 3-ary) LSDTM for subgraph G_i is convolved with the number of f_n filters with 3-ary layer by layer. Finally, (f) the enriched global geometric structure and contextual features of the p_i vertices can be obtained.

$$Q_{i,(n-1),m,k}^{n-2} = \left\{ \begin{array}{l} E_{i,(n-1),m,k}^{n-1}, E_{i,(n-2),(m-1)m+1,k}^{n-2}, \\ E_{i,(n-2),(m-1)m+2,k}^{n-2}, \dots, E_{i,(n-2),mm,k}^{n-2} \end{array} \right\} \quad (7)$$

$$E_{i,(n-1),m,k}^{n-1} = H \left(X_{i,(n-1),m,k}^{n-1}, X_{i,(n-1),m,k}^{n-1} \right) \quad (8)$$

$$E_{i,(n-2),(m-1)m+1,k}^{n-2} = H \left(X_{i,(n-1),m,k}^{n-1}, X_{i,(n-1),m,k}^{n-1} - X_{i,(n-2),(m-1)m+1,k}^{n-2} \right) \quad (9)$$

$$E_{i,(n-2),(m-1)m+2,k}^{n-2} = H \left(X_{i,(n-1),m,k}^{n-1}, X_{i,(n-1),m,k}^{n-1} - X_{i,(n-2),(m-1)m+2,k}^{n-2} \right) \quad (10)$$

⋮

$$E_{i,(n-2),mm,k}^{n-2} = H \left(X_{i,(n-1),m,k}^{n-1}, X_{i,(n-1),m,k}^{n-1} - X_{i,(n-2),mm,k}^{n-2} \right) \quad (11)$$

$$X_{i,n}^f = \xi \left[\sum_{k=1}^{f_{n-2}} \left(W_k^{n,f} \odot Q_{i,n,k}^{n-1} \right) + B^{n,f} \right] \quad (12)$$

In order to model long-range structural relationships of the point in the 3D point cloud, we analyse the structural features of the hidden layers along the (n -layer m -ary) LSDTM for G_i obtained in the previous step of graph normalisation. For the input point p_i , we construct the (n -layer m -ary) LSDTM G_i for it. Then, the GCNN is applied to extract the hidden feature information. Equation (6) defines the hidden representation $X_{i,(n-1),m}^f$. $X_{i,(n-1),m}^f$ is the f th feature of the receptive field constructed by the vertex $x_{i,(n-1),m}$ and its leaf vertices

$\{x_{i_{(n-2),(m-1)m+1}}, \dots, x_{i_{(n-2),(m-1)mm}}\}$. Here, ξ means the activation function, and Q represents the edge function, which obtains both the global shape structure and local structure information. Where, \odot is the element-wise multiplication. f_{n-2} gives the number of filters in the layer $n-2$. $W_k^{n-1,f}$ is the filter mapping from the k th to the f th in layer $n-1$. $m+1$ represents the length of filter. $B^{n-1,f}$ is the f th bias of the layer $n-1$. Here, $Q_{i_{(n-1),m,k}}^{n-2}$ is the set of directed edge functions of the m th point in the $(n-1)$ th-layer within the receptive field of $(n-2)$ th-layer. Equation (7) gives the set of directed edge functions $Q_{i_{(n-1),m,k}}^{n-2}$. Here, $E_{i_{(n-1),m,k}}^{n-1}$ shows the direct edge feature for m th point of the $(n-1)$ th-layer. The specific representation for $E_{i_{(n-1),m,k}}^{n-1}$ is as Equation (8). Similarly, Equations (9)–(11) show the representation of the $E_{i_{(n-2),(m-1)m+1,k}}^{n-2}$, $E_{i_{(n-2),(m-1)m+2,k}}^{n-2}$, $E_{i_{(n-2),(m-1)m+2,k}}^{n-2}$, and $E_{i_{(n-2),(mm),k}}^{n-2}$ edge functions respectively. In these edge functions, H concatenates the vertex feature and the relative features between the vertex and its neighbour. The $X_{i_{(n-1),m}}^f$ contains the enriched short-distance spatial structure of the m th vertice in $(n-1)$ -layer. Then, these enriched short-distance spatial structure information is propagated to long-distance by the deep GCNN along the $(n$ -layer m -ary) LSDTM for G_i .

As a result, the hidden representation $X_{i_n}^f$ can be expressed by Equation (12). It gives the f th feature channel in layer n . When $X_{i_n}^f$ is obtained, the convolution filter f_{n-1} is used to slide over the root vertice and its leaf vertices composed by hidden features to obtain the hidden feature $\tilde{X}_{i_n}^f$ for p_i , as shown in Equation (13). $\tilde{X}_{i_n}^f$ contains the enriched long-distance connections propagated by the deep GCNN. These long-distance connections contains the global spatial contiguity for p_i .

$$\tilde{X}_{i_n}^f = \xi \left[\sum_{k=1}^{f_{n-1}} \tilde{W}_k^{n,f} \odot \tilde{Q}_{i_n,k}^{n-1} + \tilde{B}^{n,f} \right] \quad (13)$$

Next, we exploit our GCNN to compute the output \tilde{O}_{i_n} . Finally, the feature vector of each point is transformed into the probability of each label through the softmax layer. The learning target of LSDTM-GCNN is to minimise the reconstruction error for the input label O_i for the vertex i and maximise the likelihood \tilde{O}_i for i . Here, cross-entropy is chosen as the loss function, which can be expressed as:

$$H_O(\tilde{O}) = - \sum_{i_n=1}^N O_{i_n} \log(\tilde{O}_{i_n}) \quad (14)$$

In this way, the LSDTM-GCNN of each central point can be learnt. In our work, we focus on applying convolution filters which slide over each root vertice and its leaf vertices of subgraph G_i to propagate the short-distance spatial structure to the long-distance connections within the subgraph G_i in a manner analogous to the standard convolution operation on

grid 2D image. These long-distance connections maintain the initial spatial contiguity, which is able to understanding enriched geometric and contextual features for the subsequent task of point clouds segmentation.

This paper implements LSDTM-GCNN for two common tasks of point cloud:

- 1) Object part segmentation. The input is a raw single object, and the output is a part category label for each point.
- 2) Semantic segmentation. The input is a 3D scan model represented by point clouds, and the output is a point-wise semantic map. Experimental results on ShapeNetPart [58], Stanford 3D Indoor Semantics (S3DIS) [59] and ScanNet [60] datasets show substantial improvements with the 3D point cloud segmentation tasks compared to the state-of-the-art.

3.4 | Analysis on differences with other methods

In the recent years, several methods [17, 18, 21, 22, 30, 33, 40, 48, 61, 62] have proposed to model the structural information between point and its contextual points. Klovov and Lempit-sky [40] design kd-trees to construct graphs to extract features from point clouds, however the construction of the graphs relies heavily on the randomised construction of trees. Different from ref. [40], the vertices of the graph in our method consider deterministically deep structured geometric relationships. Li et al. [61] propose a more generalised and flexible spectral graph convolution network, which can take raw data of different graph structures as input. It mainly transforms the graphic data in the time domain into signals in frequency domain by Fourier transform of the graph, and then performs convolution operation on the signals in the frequency domain. However, the signal transformation from spatial to spectral domains and vice-versa results in computational complexity $O(n^2)$.

Wang et al. [17] exploit the local relationships between points in a point cloud via the graph pointnet module. And Zhang et al. [18] delicately design local k-NNs patterns to capture both global and local spatial layout of point clouds. Chen et al. [30] learn local geometric representations by embedding graph attention mechanisms in the MLP layer. Hegde and Gangisetty [21] propose to use the Inception module to replace the mlp module in pointnet to extract local features to capture fine-grained details, and then utilises GAP for aggregation to obtain global features. Compared to the max pooling layer, the GAP layer not only acquires global features, but also provides a more native way for the convolutional structure to enforce the correspondence between features and point cloud parts. Refs. [17, 18, 21, 30] focus on exploring local feature relationships of point clouds, and then merely use local aggregation to obtain global features. A common limitation of these methods is that they do not simultaneously take into account fine local details and long-range contextual information. Engel et al. [33] design a self-attention network to

aggregate local features, and subtraction relation to generate the attention weights for 3D point cloud. However, it is still limited in directly acquiring long-range contexts. In addition, when self-attention is used to acquire deeper local neighbourhoods, its computational complexity will be greatly increased. Similar to ref. [33], Wu et al. [48] propose to utilise Transformer-based self-attention to learn long-range pairwise relationships. In GraphTrans, the input information is first input to MPNNs for processing, and then input to Transformer, but MPNNs are limited by problems of over-smoothing, over-squashing, and low expressivity against the WL test [63]. Therefore, some early information may not be well preserved, resulting in the loss of important information. Qian et al. [22] design residual connections, inverse bottleneck design and separable MLP to effectively extend the pointnet++ model. PointNeXt XL achieves state-of-the-art performance in 3D point cloud semantic segmentation. However, it is more computationally expensive in terms of triggers, and moreover, it contains more than twice the number of parametric layers compared to our model. Wijaya et al. [62] utilise multiple residual blocks and multiple learnable pooling to learn high-semantic and high-resolution features of point clouds. But PointStack suffers from similar failures as PointNeXt XL, that is, it contains more training parameters and requires expensive computation. Besides, when the number of training samples is insufficient, its performance drops drastically. Unlike these methods, in order to capture structural relationships over long distances, our subgraphs can build the short-distance structural relationship between a point and its contextual points and propagate to the long range through the deep GCNN. With the exploitation of deeply discovered inter-dependencies relations, our approach is able to understand 3D scene more comprehensively.

4 | EXPERIMENTS

4.1 | Datasets and evaluation metrics

Our experiments are mainly conducted on three widely used datasets: ShapeNetPart, Stanford 3D Indoor Semantics Dataset (S3DIS), and ScanNet. ShapeNetPart contains 16,881 3D shapes from 16 categories, annotated with 50 parts in total. Each point sampled from shapes is assigned with one of the 50 different parts. The S3DIS dataset covers six large-scale indoor areas from three different buildings for a total of 271 scenes captured from three different buildings. Each point cloud is a medium-sized single room ($20 \times 15 \times 5$ m) with dense 3D points. The input is a complete scene point cloud generated using dense RGB-D mapping. Each point in the scene point cloud is associated with an instance label and a semantic label from 13 categories. To prepare the training data, following ref. [13] where the points are uniformly sampled into blocks of area size 1×1 m. During the training, we randomly sample 4096 points from each block on-the-fly. In our research, to evaluate overall segmentation accuracy, we utilise the standard

6-fold cross validation in our experiment. ScanNet dataset contains richly annotated RGB-D scans of real-world environments containing 2.5M RGB-D images in 1513 scans acquired in 707 distinct spaces.

For the evaluation metrics, we employ mean of class-wise intersection over union (mIoU) and overall accuracy (OA) following ref. [12].

4.2 | Object part segmentation on ShapeNetPart

Following the setting in ref. [13], point clouds are generated by uniformly sampling shapes from ShapeNet. Taken shapes represented by point clouds as input, the task is to predict a part label for each point. We employ the official train test split following ref. [58]. This experiment is conducted as a toy example to demonstrate the effectiveness of our approach on semantic part segmentation for point clouds.

Table 1 shows the results of different methods evaluated regarding mIoU in each category. For fair comparison, the input of these methods removes the normal information. We only use the coordinate information as input. In this highly competitive dataset, our LSDTM-GCNN achieves on par performance with most methods in the metric mIoU. There is still a small gap between our LSDTM-GCNN and refs. [21, 23, 27, 33, 62]. Li et al. [23] utilise x -transformation from the input points to permute unordered local points into a latent potentially canonical order. Thomas et al. [27] design an MLP or discrete kernel points to simulate a local continuous convolution kernel of the centre points. Refs. [23, 27] can capture more details of a 3D point cloud, but they both require more than twice the number of parametric layers and times as required by our LSDTM-GCNN to achieve the reported performance. Although Engel et al. [33] achieve competitive results with the state-of-the-art, it adopts an attention mechanism to aggregate local features, which greatly increases the computational complexity. Wijaya et al. [62] attempt to use multiple residual blocks and multiple learnable pooling to obtain high-semantic information. Similar to ref. [33], PointStack also needs more training parameters and expensive computation. Hegde and Gangisetty [21] propose to use GAP instead of maxpooling to obtain global features. Most of the above methods do not directly take into account long-range relationships of 3D point clouds, limiting their ability to understand 3D scene more comprehensively. Compared to maxpooling, GAP [68] can better preserve the spatial location information and make it have a global receptive field. When we use GAP to obtain global features of short-distance subgraph, the more effective short-range spatial structure is propagated to longer distances through GCNN to discover the underlying relationships. Experimental results show that our LSDTM-GCNN (GAP) achieves the same excellent mIoU with ref. [33]. This phenomenon indicates that long short-distance topology structure information of the points in the 3D point cloud is essential for understanding the semantics of a point cloud.

TABLE 1 Object part segmentation results on ShapeNetPart

| Methods | mIoU | Aero | Bag | Cap | Car | Chair | Ear phone | Guitar | Knife | Lamp | Laptop | Motor | Mug | Pistol | Rocket | Skate board | Table |
|-------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|-------------|
| 3DCNN [12] | 79.4 | 75.1 | 72.8 | 73.3 | 70.0 | 87.2 | 63.5 | 88.4 | 79.6 | 74.4 | 93.9 | 58.7 | 91.8 | 76.4 | 51.2 | 65.3 | 77.1 |
| O-CNN [64] | 81.4 | 81.0 | 78.4 | 77.7 | 75.7 | 87.6 | 61.9 | 92.0 | 85.4 | 82.5 | 95.7 | 70.6 | 91.9 | 85.9 | 53.1 | 69.8 | 75.3 |
| Kd-network [40] | 82.3 | 80.1 | 74.6 | 74.3 | 70.3 | 88.6 | 73.5 | 90.2 | 87.2 | 81.0 | 94.9 | 57.4 | 86.7 | 78.1 | 51.8 | 69.9 | 80.3 |
| PointNet [12] | 83.7 | 83.4 | 78.7 | 82.5 | 74.9 | 89.6 | 73.0 | 91.5 | 85.9 | 80.8 | 95.3 | 65.2 | 93.0 | 81.2 | 57.9 | 72.8 | 80.6 |
| Synspeccnn [39] | 84.7 | 81.6 | 81.7 | 81.9 | 75.2 | 90.2 | 74.9 | 93.0 | 86.1 | 84.7 | 95.6 | 66.7 | 92.7 | 81.6 | 60.6 | 82.9 | 82.1 |
| Kc-Net [65] | 84.7 | 82.8 | 81.5 | 86.4 | 77.6 | 90.3 | 76.8 | 91.0 | 87.2 | 84.5 | 95.5 | 69.2 | 94.4 | 81.6 | 60.1 | 75.2 | 81.3 |
| So-Net [14] | 84.9 | 82.8 | 77.8 | 88.0 | 77.3 | 90.6 | 73.5 | 90.7 | 83.9 | 82.8 | 94.8 | 69.1 | 94.2 | 80.9 | 53.1 | 72.9 | 83.0 |
| PointNet++ [13] | 84.9 | 82.9 | 81.3 | 85.3 | 78.5 | 90.6 | 73.3 | 91.2 | 86.9 | 82.5 | 95.3 | 71.8 | 95.0 | 82.1 | 57.7 | 75.4 | 81.6 |
| ASIS [34] | 85.0 | 81.4 | 81.2 | 86.1 | 72.3 | 90.4 | 74.1 | 91.7 | 85.6 | 83.0 | 95.7 | 70.4 | 94.6 | 81.4 | 53.2 | 78.1 | 82.0 |
| Weak.Sup [56] | 85.0 | 83.1 | 82.6 | 80.8 | 77.7 | 90.4 | 77.3 | 90.9 | 87.6 | 82.9 | 95.8 | 64.7 | 93.9 | 79.8 | 61.9 | 74.9 | 82.9 |
| SK-Net [66] | 85.0 | 82.9 | 80.7 | 87.6 | 77.8 | 90.5 | 79.9 | 91.0 | 88.1 | 84.0 | 95.7 | 69.9 | 94.0 | 81.1 | 60.8 | 76.4 | 81.9 |
| SpiderCNN [25] | 85.3 | 83.5 | 81.0 | 87.2 | 77.5 | 90.7 | 76.8 | 91.1 | 87.3 | 83.3 | 95.8 | 70.2 | 93.5 | 82.7 | 59.7 | 75.8 | 82.8 |
| SRN [67] | 85.3 | 82.4 | 79.8 | 88.1 | 77.9 | 90.7 | 69.6 | 90.9 | 86.3 | 84.0 | 95.4 | 72.2 | 94.9 | 81.3 | 62.1 | 75.9 | 83.2 |
| LKPO-GNN [18] ($k = 8$) | 85.3 | 82.5 | 81.8 | 87.7 | 78.8 | 90.7 | 75.4 | 90.8 | 87.1 | 83.5 | 95.6 | 72.0 | 95.6 | 81.7 | 55.9 | 75.8 | 82.8 |
| LKPO-GNN [18] ($k = 32$) | 85.6 | 82.6 | 80.8 | 86.9 | 78.6 | 90.9 | 77.7 | 90.8 | 86.9 | 84.9 | 95.8 | 71.7 | 94.6 | 82.4 | 56.1 | 76.0 | 82.8 |
| PIG-Net [21] | 85.9 | 84.2 | 83.1 | 88.9 | 78.6 | 91.7 | 78.2 | 94.4 | 89.5 | 94.2 | 96.3 | 66.2 | 91.6 | 85.1 | 64.8 | 93.5 | 94.2 |
| PointCNN [23] | 86.1 | 84.1 | 86.5 | 86.0 | 80.8 | 90.6 | 79.7 | 92.3 | 88.4 | 85.3 | 95.1 | 77.2 | 95.3 | 84.2 | 64.2 | 80.0 | 83.0 |
| KPConv [27] | 86.2 | 83.6 | 86.7 | 87.2 | 79.1 | 89.1 | 77.8 | 92.6 | 88.4 | 82.7 | 96.2 | 78.1 | 95.8 | 85.4 | 69.0 | 82.0 | 83.6 |
| PointStack [62] | 86.4 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Point transformer [33] | 86.6 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| LSDTM-GCNN | 85.8 | 82.2 | 81.2 | 90.3 | 72.8 | 91.4 | 78.6 | 89.3 | 87.9 | 79.8 | 96.4 | 62.6 | 94.7 | 79.6 | 60.6 | 75.1 | 86.9 |
| LSDTM-GCNN (GAP) | 86.6 | 84.4 | 86.6 | 90.2 | 79.9 | 91.9 | 80.6 | 95.3 | 90.0 | 83.2 | 96.1 | 71.7 | 95.2 | 80.3 | 68.1 | 79.3 | 97.2 |

Note: Bold values represent the best results.

Abbreviations: CNN, convolutional neural network; GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled; mIoU, intersection over union.

4.3 | S3DIS semantic segmentation

In this section, we perform experiments on S3DIS dataset to evaluate our performance on large real scene scans. Following experimental settings in RandLA-Net [19], we split the dataset into training and testing sets. We report the result on six-fold cross-validation calculating the metrics with results from different folds merged. For the evaluation metrics, we utilise mIoU and OA. Each point is represented by 9D vector (XYZ, RGB and normalised location as to the room). In each block, we uniformly sample 4096×3 points as input. Here, in order to reflect the long short-distance contextual information of the point cloud, we only utilise 3D coordinates of points. At test time, we test on all the points.

As shown in Table 2, our LSDTM-GCNN achieves better performance than [20, 22, 27, 33] results of other studies. For fair comparison, these methods all take coordinate points as

input. It is worth noting that, most methods are limited to exploring the underlying relationships and interactions of contextual points in 3D space. By contrast, our LSDTM-GCNN interpret implicit contextual-point interaction and long-distance contextual relationship using a general representation. Although ref. [33] performs slightly better than our model, it is limited in its ability to directly capture long-range features. In additional, we present the methods using the positions (X, Y, Z) and RGB as additional features input in Table 3. Our LSDTM-GCNN significantly outperforms most other methods in OA. This indicates that the proposed LSDTM-GCNN is significant and effective for improving the semantic segmentation performance. Compared to ref. [22], our model is slightly worse in mIoU. This is due to the fact that an inverted residual bottleneck separable MLPs extracting enriched features. However, it contains more parameters and requires higher computational cost.

TABLE 2 Segmentation results on S3DIS with (X,Y,Z) (6-fold cross validation)

| Methods | OA | mIoU | Ceiling | Floor | Wall | Beam | Column | Window | Door | Chair | Table | Bookcase | Sofa | Board | Clutter |
|------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| RSNet [28] | - | 51.9 | 93.3 | 98.3 | 79.1 | 0.00 | 15.7 | 45.4 | 50.1 | 65.5 | 67.9 | 22.5 | 52.5 | 41.0 | 43.6 |
| PointNet [12] | 80.6 | 54.8 | 87.9 | 97.1 | 65.1 | 50.0 | 41.1 | 63.9 | 47.4 | 63.9 | 65.4 | 29.9 | 44.8 | 9.6 | 46.4 |
| DGCNN [41] | 83.9 | 55.8 | 87.5 | 91.3 | 61.9 | 1.0 | 19.3 | 55.4 | 31.3 | 68.6 | 71.9 | 19.1 | 48.7 | 38.9 | 51.3 |
| ASIS [34] | 86.2 | 59.3 | 91.3 | 89.7 | 69.8 | 45.8 | 27.0 | 51.9 | 55.1 | 61.0 | 49.3 | 9.1 | 40.2 | 33.5 | 40.7 |
| SPG [42] | 85.5 | 62.1 | 89.9 | 95.1 | 76.4 | 62.8 | 47.1 | 55.3 | 68.4 | 73.5 | 69.2 | 63.2 | 45.9 | 8.7 | 52.9 |
| PointCNN [23] | - | 62.7 | 85.6 | 85.2 | 77.1 | 63.8 | 34.8 | 56.1 | 69.3 | 60.8 | 71.2 | 64.3 | 43.2 | 47.9 | 56.3 |
| PointSIFT [16] | 83.6 | 63.8 | 86.5 | 86.3 | 71.9 | 54.5 | 30.0 | 65.4 | 66.3 | 64.6 | 77.9 | 52.1 | 53.7 | 58.8 | 61.6 |
| RandLA-Net [19] | 84.2 | 63.2 | 91.5 | 96.4 | 72.9 | 61.4 | 46.3 | 49.7 | 58.9 | 69.7 | 58.8 | 66.6 | 62.4 | 36.4 | 50.0 |
| LKPO-GNN [18] | 85.8 | 64.6 | 83.4 | 85.6 | 73.1 | 63.3 | 36.7 | 64.4 | 70.5 | 65.9 | 79.1 | 46.3 | 54.5 | 55.9 | 61.7 |
| KPConv [27] | - | 65.3 | 91.7 | 93.9 | 80.8 | 63.4 | 49.7 | 60.4 | 70.9 | 68.1 | 69.6 | 53.4 | 58.6 | 51.1 | 58.3 |
| SCF-Net [20] | 86.1 | 65.8 | 91.9 | 94.6 | 75.6 | 65.6 | 45.2 | 50.9 | 59.0 | 69.4 | 79.3 | 60.6 | 61.7 | 39.2 | 53.8 |
| PointNeXt-XL [22] | 86.0 | 66.1 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Point transformer [33] | 90.2 | 73.5 | 94.3 | 97.5 | 84.7 | 55.6 | 58.1 | 66.1 | 78.2 | 77.6 | 74.1 | 67.3 | 71.2 | 65.7 | 64.8 |
| LSDTM-GCNN | 86.6 | 66.5 | 92.9 | 95.7 | 74.0 | 61.4 | 54.3 | 67.8 | 66.1 | 68.7 | 74.5 | 49.0 | 57.0 | 41.2 | 61.0 |
| LSDTM-GCNN (GAP) | 88.1 | 71.6 | 93.7 | 95.9 | 85.2 | 60.9 | 61.3 | 66.9 | 72.4 | 72.5 | 78.0 | 52.7 | 64.3 | 62.2 | 65.3 |

Note: Bold values represent the best results.

Abbreviations: LSDTM, long short-distance topologically modelled; mIOU, intersection over union; OA, overall accuracy; S3DIS, Stanford 3D indoor semantics.

TABLE 3 Segmentation results on S3DIS with (X,Y,Z,R,G,B) (6-fold cross validation)

| Methods | OA | mIoU | Ceiling | Floor | Wall | Beam | Column | Window | Door | Chair | Table | Bookcase | Sofa | Board | Clutter |
|-------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| PointNet [12] | 78.6 | 47.6 | 88.0 | 88.7 | 69.3 | 42.4 | 23.1 | 47.5 | 51.6 | 54.1 | 42.0 | 9.6 | 38.2 | 29.4 | 35.2 |
| RSNet [28] | - | 56.5 | 92.5 | 92.8 | 78.6 | 32.8 | 34.4 | 51.6 | 68.1 | 59.7 | 60.1 | 16.4 | 50.2 | 44.9 | 52.0 |
| DGCNN [41] | 83.9 | 55.8 | 87.5 | 91.3 | 61.9 | 1.0 | 19.3 | 55.4 | 31.3 | 68.6 | 71.9 | 19.1 | 48.7 | 38.9 | 51.3 |
| SPG [42] | 86.4 | 62.1 | 89.9 | 95.1 | 76.4 | 62.8 | 47.1 | 55.3 | 68.4 | 73.5 | 69.2 | 63.2 | 45.9 | 8.7 | 52.9 |
| PointCNN [23] | 88.1 | 65.4 | 94.8 | 97.3 | 75.8 | 63.3 | 51.7 | 58.4 | 57.2 | 71.6 | 69.1 | 39.1 | 61.2 | 52.2 | 58.6 |
| ELGS [17] | 87.6 | 66.3 | 93.7 | 95.6 | 76.9 | 42.6 | 46.7 | 63.9 | 69.0 | 70.1 | 76.0 | 52.8 | 57.2 | 54.8 | 62.5 |
| PointWeb [15] | 87.3 | 66.7 | 93.5 | 94.2 | 80.8 | 52.4 | 41.3 | 64.9 | 68.1 | 71.4 | 67.1 | 50.3 | 62.7 | 62.2 | 58.5 |
| ShellNet [69] | 87.1 | 66.8 | 90.2 | 93.6 | 79.9 | 60.4 | 44.1 | 64.9 | 52.9 | 71.6 | 84.7 | 53.8 | 64.6 | 48.6 | 59.4 |
| RandLA-Net [19] | 88.0 | 70.0 | 93.1 | 96.1 | 80.6 | 62.4 | 48.0 | 64.4 | 69.4 | 69.4 | 76.4 | 60.0 | 64.2 | 65.9 | 60.0 |
| KPConv [27] | - | 70.6 | 93.6 | 92.4 | 83.1 | 63.9 | 54.3 | 66.1 | 76.6 | 57.8 | 64.0 | 69.3 | 74.9 | 61.3 | 60.3 |
| SCF-Net [20] | 88.4 | 71.6 | 93.3 | 96.4 | 80.9 | 64.9 | 47.4 | 64.5 | 70.1 | 71.4 | 81.6 | 67.2 | 64.4 | 67.5 | 60.9 |
| AF-Net [70] | 88.9 | 72.2 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| PointNeXt-XL [22] | 90.3 | 74.9 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| LSDTM-GCNN | 89.3 | 72.4 | 94.8 | 97.5 | 79.8 | 60.6 | 54.0 | 80.4 | 78.4 | 75.0 | 77.0 | 50.8 | 58.8 | 68.7 | 65.8 |
| LSDTM-GCNN (GAP) | 90.5 | 74.6 | 95.3 | 95.7 | 81.6 | 63.3 | 55.1 | 69.4 | 80.1 | 76.5 | 78.2 | 70.3 | 62.5 | 71.7 | 69.6 |

Note: Bold values represent the best results.

Abbreviations: LSDTM, long short-distance topologically modelled; mIOU, intersection over union; OA, overall accuracy; S3DIS, Stanford 3D indoor semantics.

From top to bottom, Figure 3 shows visualisation examples of five typical indoor scenes, including conference room, openspace, lounge, office and copyRoom. We compare our method with the RandLA-Net, SCF-Net and AF-Net methods, and the predictions of our LSDTM-GCNN are closer to the

ground truth in conference room, lounge, office and copyRoom scenes. We can see that the LSDTM-GCNN captures certain detailed structures in the point clouds well. As shown in the Figure 3, inconspicuous object parts, like legs of chair and table, can be distinguished and recognised correctly. Nevertheless, the

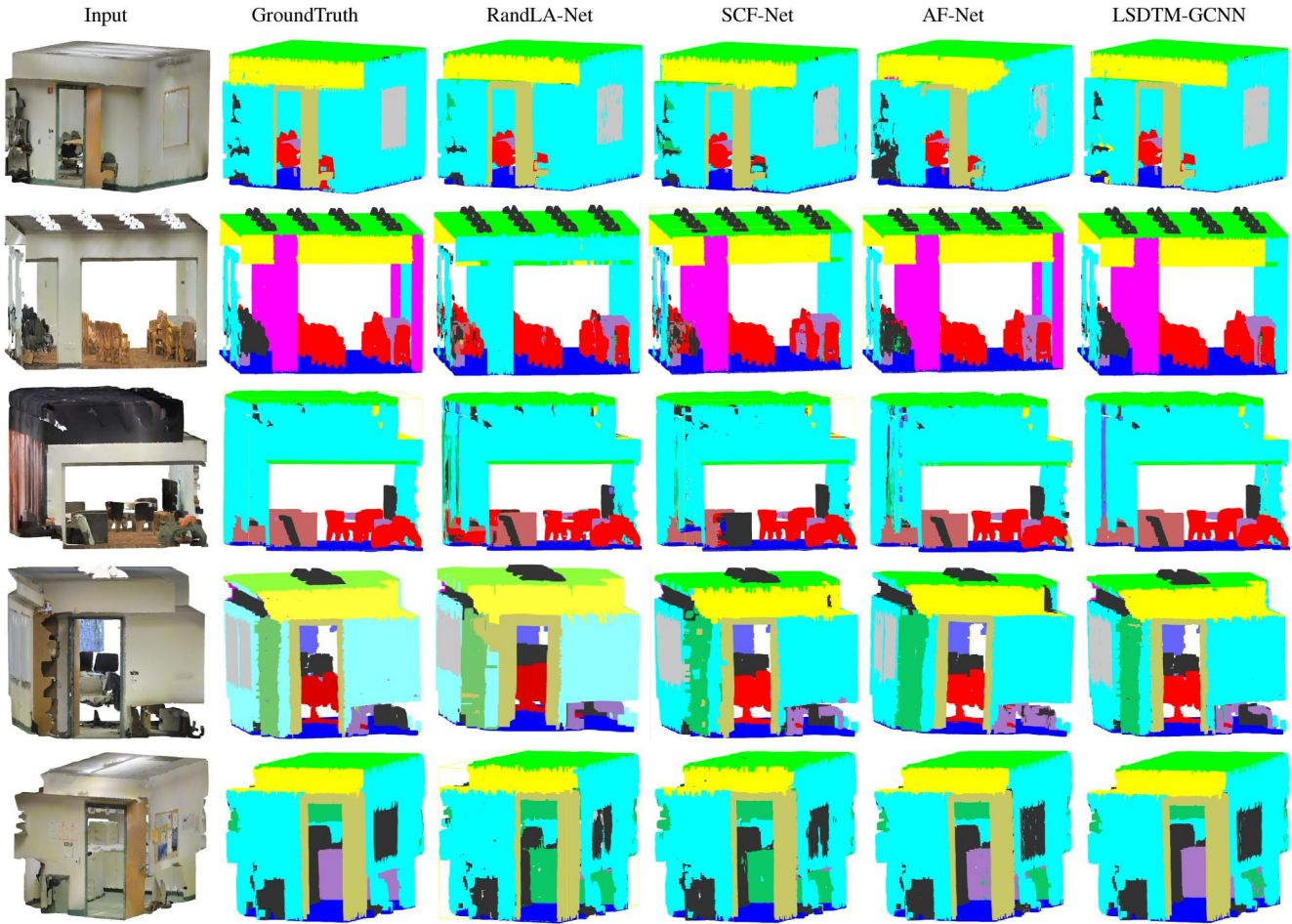


FIGURE 3 Visualisation of semantic segmentation results on S3DIS dataset. GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled

misclassification is inevitable. As shown in the copyRoom of Figure 3, the clutter (part of the area) is misclassified to bookcase.

4.4 | Scannet semantic segmentation

Following experimental settings in Grid-GCN [57], we split the ScanNet dataset into 1201 training scenes and 312 testing scenes sets. For semantic segmentation in ScanNet dataset, each shape is represented by a 3D point cloud with 8192 points, as in ref. [52]. In this section, the experiment is implemented using Tensorflow toolbox and models are trained on four GeForce RTX 3090.

In Table 4, we present the OA (%) and mIoU (%) scores. This experiment demonstrates the effectiveness of our approach on semantic segmentation for 3D point cloud. We compare our method with previous state-of-the-art method [57] in Table 4. Ref. [57] consumes CAGQ to achieve efficient data structuring and computation, and achieved better performance on point cloud segmentation. However, it only constructs local graphs based on points and their neighbours.

The way of constructing graphs is limited to discover the underlying relationships between the points in point cloud.

4.5 | Ablation study

The goal of our ablation study is to evaluate the effectiveness of each component of our LSDTM-GCNN module.

All ablated networks are trained with the same network settings and test on S3DIS dataset. We report the result using six-fold cross-validation. The OA and mIoU are exploited as the evaluation metrics.

- 1) Remove LSDTM-GCNN module. This baseline disables the discovery of the long-distance structural interactions between a point and its contextual points. After removing LSDTM-GCNN, the point features are directly fed into the subsequent max-pooling.
- 2) Replace $E_i(f_i, \text{knn}(p_i))$ with $E_i(f_i)$. In this baseline, the edge function $E_i(f_i)$ only considers the global information and ignores the local contextual relationship between points and its neighbours.

- 3) Replace $E_i(f_i, \text{knn}(p_i))$ with $E_i(\text{knn}(p_i))$. Note such a choice only encodes the local shape information and ignores the global shape information. Similar to baseline 2), we utilise local features to replace the coordinates of the points for global feature learning.
- 4) Replace 3-level 3-ary with 2-level 3-ary. For better understanding of effect of LSDTM-GCNN architectures for smaller depth, we conducted the experiment corresponding to 2-level 3-ary subgraph.
- 5) Replace 3-level 3-ary with 3-level 2-ary. For better understanding of effect of LSDTM-GCNN architectures for smaller width, we conducted the experiment corresponding to 3-level 2-ary subgraph.
- 6) Randomised construction 3-level 3-ary. To demonstrate the validity of deterministic construction of subgraph, we construct the 3-level 3-ary subgraph with random sampling.

Table 5 shows the OA and mIoU scores of all ablated baselines. From this ablation study, we can see our proposed LSDTM-GNN modules are complementary with each other to obtain the state-of-the-art performance. In addition, we also observed that the removal of LSDTM-GNN module causes the second jump in term of performance.

The case study of this module is highlighted in Figure 2, which shows the effectiveness of the triple-layer ternary topology to obtain enriched contextual information of points with spatial contiguity achieved crossover the 3D space. These short and long range contextual information plays crucial roles in 3D point cloud segmentation and semantic understanding. In the second experiment, we remove local contextual relationship $\text{knn}(p_i)$, but keep global geometric shape structure f_i in the edge function E_i . Our experimental results show that removing the $\text{knn}(p_i)$ module diminishes performance due to the negligence in local contextual information modelling. In the third experiment, the removal of the

global shape structure f_i unit diminishes performance by not being able to effectively retain the spatial contiguity. In the fourth experiment, in order to explore the effect of the smaller depth on the LSDTM-GCNN architectures, we replace 3-level 3-ary with 2-level 3-ary. Replacing 3-level 3-ary with 2-level 3-ary unit shows the third greatest impact on performance, demonstrating the effectiveness of smaller depth in preservation of the long range contextual information. In a fifth experiment, we replace 3-level 3-ary with 3-level 2-ary unit. The results show the fourth greatest impact on performance, illustrating the effectiveness of smaller width in obtaining the short-distance contextual information. Finally, to verify the effectiveness of deterministic structural subgraph, we replace 3-level 3-ary LSDTM subgraph with randomised construction 3-level 3-ary subgraph. Randomised construction 3-level 3-ary subgraph discards the initial spatial contiguity, which makes its experimental results the worst than 3-level 3-ary LSDTM subgraph. This ablation study shows LSDTM units complement each other to achieve the state-of-the-art performance.

4.6 | Time and space complexity

Table 6 summarises time (float-point operations) and space (number of parameters in the work) complexity of our object part segmentation on ShapeNetPart. We record the inference times with a batch size of 1 using TensorFlow on a single GeForce RTX 3090. We also compare LSDTM-GCNN with previous research. In terms of space complexity, ref. [13] has smallest model size, but it did not consider the structural interactions between contextual points in 3D point cloud. Compared with refs. [14, 18, 21, 23, 27, 62, 65], which mostly consider the structure information, our model achieved acceptable model size. In terms of time complexity, LSDTM-GCNN (GAP) shows the fastest forward time. Although Ref. [18] ($k = 8$) achieves faster inference time than our LSDTM-GCNN, our LSDTM-GCNN and LSDTM-GCNN (GAP) outperform [18] by 0.5 and 1.3 in mIoU metric respectively. Zhang et al. [18] are unable to simultaneously take into account fine local details and long-range contextual

TABLE 4 Semantic segmentation on ScanNet

| Methods | OA | mIoU |
|------------------|-------------|-------------|
| 3DCNN [71] | 73.0 | - |
| PointNet [12] | 73.9 | - |
| SK-Net [66] | 81.4 | - |
| PointNet++ [13] | 84.0 | 56.9 |
| PointCNN [23] | 84.8 | - |
| PointSIFT [16] | 85.1 | 54.5 |
| ELGS [17] | 85.3 | 40.6 |
| LKPO-GNN [18] | 85.3 | 58.4 |
| Grid-GCN [57] | 85.4 | - |
| SegGCN [46] | - | 58.9 |
| LSDTM-GCNN | 85.4 | 59.1 |
| LSDTM-GCNN (GAP) | 86.1 | 59.7 |

Note: Bold values represent the best results.

Abbreviations: GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled; mIoU, intersection over union; OA, overall accuracy.

TABLE 5 The mIoU scores of all ablated networks based on full LSDTM-GCNN with (X,Y,Z)

| Baselines | OA | mIoU |
|--|-------------|-------------|
| 1) Remove LSDTM-GCNN module | 85.3 | 59.6 |
| 2) Replace $E_i(f_i, \text{knn}(p_i))$ with $E_i(f_i)$ | 85.9 | 66.2 |
| 3) Replace $E_i(f_i, \text{knn}(p_i))$ with $E_i(\text{knn}(p_i))$ | 85.7 | 65.2 |
| 4) Replace 3-level 3-ary with 2-level 3-ary | 85.7 | 63.7 |
| 5) Replace 3-level 3-ary with 3-level 2-ary | 85.8 | 64.9 |
| 6) Randomised construction 3-level 3-ary | 84.6 | 57.1 |
| 7) The full framework (LSDTM-GCNN) | 86.6 | 66.5 |

Note: Bold values represent the best results.

Abbreviations: GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled; mIoU, intersection over union; OA, overall accuracy.

TABLE 6 The complexity comparison

| Methods | Forward times | Model size |
|----------------------------|---------------|-------------|
| PointNet [12] | 128.26 | 98.2 |
| Kc-Net [65] | 36.5 | 25.7 |
| So-Net [14] | 68.3 | 59.5 |
| PointNet++ [13] | 24.63 | 16.2 |
| LKPO-GNN [18] ($k = 8$) | 19.94 | 28.2 |
| LKPO-GNN [18] ($k = 32$) | 27.64 | 51.1 |
| PointCNN [23] | 80.43 | 95.4 |
| KPConv [27] | 70.4 | 58.8 |
| PIG-Net [21] | 84.8 | 81.4 |
| PointStack [62] | 91.2 | 139.7 |
| LSDTM-GCNN | 22.5 | 23.5 |
| LSDTM-GCNN (GAP) | 17.6 | 19.3 |

Note: Bold values represent the best results.

Abbreviations: CNN, convolutional neural network; GCNN, graph convolutional neural network; LSDTM, long short-distance topologically modelled.

information, which is desired for point clouds. This phenomenon shows that our model has great potential for real-time applications.

5 | CONCLUSION

This paper presents an effective plug-and-play module called LSDTM-GCNN for point cloud segmentation and understanding. LSDTM-GCNN is a deterministic graph-like topology that formulates the inner-points interactions and semantic contexts by building short-distance connected subgraphs and propagating the connections to long-distance over the deep GCNN. Our proposed method achieves enriched structural and contextual feature representation with the global spatial contiguity for 3D point cloud. Experiments show that our algorithm achieves promising performance on 3D point cloud segmentation, which can be a new baseline for point cloud segmentation and semantic understanding benchmarks.

Our method is an intuitive, simple yet flexible framework for 3D point cloud representation and parsing. As the first trial of using subgraph for point clouds, there are still rooms for improvement. The future work will investigate: (1) accelerating graph construction by reducing the query scope of nearest neighbour. (2) exploring more advanced sampling strategy of centre points to improve coverage.

AUTHOR CONTRIBUTIONS

Wen Jing Zhang: Formal analysis; Methodology; Software; Validation; Visualisation; Writing – original draft; Writing – review & editing. **Song Zhi Su:** Formal analysis; Supervision; Writing – review & editing. **Qing Qi Hong:** Formal analysis; Supervision. **Bei Zhan Wang:** Formal analysis; Supervision. **Li Sun:** Formal analysis; Supervision; Writing – review & editing.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

We declare that we have no conflict of interest.

DATA AVAILABILITY STATEMENT

For ShapeNetPart, please refer to http://web.stanford.edu/~ericji/project_page/part_annotation/index.html. For S3DIS, please refer to <http://buildingparser.stanford.edu/dataset.html%23Download>. For Scannet, please refer to <http://www.scan-net.org/>.

ORCID

Song Zhi Su  <https://orcid.org/0000-0001-8961-9405>

REFERENCES

1. Jason, K., et al.: Joint 3D proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1–8 (2018)
2. Liu, Z., et al.: Real-time 6D Lidar SLAM in large scale natural terrains for UGV. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 662–667 (2018)
3. Aytaylan, H., Yuksel, S.E.: Fully-connected semantic segmentation of hyperspectral and LiDAR data. *IET Comput. Vis.* 13(3), 285–293 (2019). <https://doi.org/10.1049/iet-cvi.2018.5067>
4. Yan, L., et al.: RTL3D: real-time LiDAR-based 3D object detection with sparse CNN. *IET Comput. Vis.* 14(5), 224–232 (2020). <https://doi.org/10.1049/iet-cvi.2019.0508>
5. Li, X., Liu, J., Dai, S.: Point cloud super-resolution based on geometric constraints. *IET Comput. Vis.* 15(4), 312–321 (2021). [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi.2021.2045>
6. Tan, J., et al.: SASO: joint 3D semantic-instance segmentation via multi-scale semantic association and salient point clustering optimization. *IET Comput. Vis.* 15(5), 366–379 (2021). [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi.2021.2033>
7. Tchapmi, L., et al.: SEGCloud: semantic segmentation of 3D point clouds. In: 2017 International Conference on 3D Vision (3DV), pp. 537–547 (2017)
8. Wang, W., et al.: Shape inpainting using 3D generative adversarial network and recurrent convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2298–2306 (2017)
9. Qi, C.R., et al.: Volumetric and multi-view CNNs for object classification on 3D data. In: Computer Vision and Pattern Recognition, pp. 5648–5656 (2016)
10. Lawin, F.J., et al.: Deep projective 3D semantic segmentation. In: International Conference on Computer Analysis of Images and Patterns, pp. 95–107 (2017)
11. Yang, B., Luo, W., Urtasun, R.: PIXOR: real-time 3D object detection from point clouds. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7652–7660 (2018)
12. Charles, R.Q., et al.: PointNet: deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85 (2017)
13. Qi, C.R., et al.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: arXiv: Computer Vision and Pattern Recognition, pp. 5105–5114 (2017)
14. Li, J., Chen, B.M., Lee, G.H.: So-Net: self-organizing network for point cloud analysis. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9397–9406 (2018)
15. Zhao, H., et al.: PointWeb: enhancing local neighborhood features for point cloud processing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5560–5568 (2019)

16. Jiang, M., Wu, Y., Lu, C.: PointSIFT: a SIFT-like network module for 3D point cloud semantic segmentation. In: *Computer Vision and Pattern Recognition* (2018)
17. Wang, X., He, J., Ma, L.: Exploiting local and global structure for point cloud semantic segmentation with contextual point representations. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, December 8–14, 2019, Vancouver, BC, Canada, pp. 4573–4583 (2019)
18. Zhang, W., et al.: Local k-NNs pattern in omni-direction graph convolution neural network for 3D point clouds. *Neurocomputing* 413, 487–498 (2020). <https://doi.org/10.1016/j.neucom.2020.06.095>
19. Hu, Q., et al.: RandLA-Net: efficient semantic segmentation of large-scale point clouds. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11105–11114 (2020)
20. Fan, S., et al.: SCF-Net: learning spatial contextual features for large-scale point cloud segmentation. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14499–14508 (2021)
21. Hegde, S., Gangisetty, S.: PIG-Net: inception based deep learning architecture for 3D point cloud segmentation. *Comput. Graph.* 95, 13–22 (2021). <https://doi.org/10.1016/j.cag.2021.01.004>
22. Qian, G., et al.: PointNeXt: revisiting pointnet++ with improved training and scaling strategies. In: *Computer Vision and Pattern Recognition* (2022). [Online]. <https://doi.org/10.48550/arXiv.2206.04670>
23. Li, Y., et al.: PointCNN: convolution on x-transformed points. *Adv. Neural Inf. Process. Syst.* 31, 1672 (2018)
24. Hua, B.-S., Tran, M.-K., Yeung, S.-K.: Pointwise convolutional neural networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 984–993 (2018)
25. Yifan, X., et al.: SpiderCNN: deep learning on point sets with parameterized convolutional filters. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 90–105 (2018)
26. Komarichev, A., Zhong, Z., Hua, J.: A-CNN: annularly convolutional neural networks on point clouds. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 06, pp. 7413–7422 (2019)
27. Thomas, H., et al.: KPConv: flexible and deformable convolution for point clouds. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6410–6419 (2019)
28. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3D segmentation of point clouds. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2626–2635 (2018)
29. Ye, X., et al.: 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In: *Computer Vision – ECCV 2018*, pp. 415–430 (2018)
30. Chen, C., Fragonara, L.Z., Tsoyros, A.: GAPointNet: graph attention based point neural network for exploiting local feature of point cloud. *Neurocomputing* 438, 122–132 (2021). <https://doi.org/10.1016/j.neucom.2021.01.095>
31. Wang, L., et al.: Graph attention convolution for point cloud semantic segmentation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
32. Cheng, R., et al.: (AF)²-S3Net: attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12542–12551 (2021)
33. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer. *IEEE Access* 9, 134826–134840 (2021). <https://doi.org/10.1109/access.2021.3116304>
34. Wang, X., et al.: Associatively segmenting instances and semantics in point clouds. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4091–4100 (2019)
35. Pham, Q.-H., et al.: JSIS3D: joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8819–8828 (2019)
36. Zhao, L., Tao, W.: JSNet: joint instance and semantic segmentation of 3D point clouds. *Proc. AAAI Conf. Artif. Intell.* 34(07), 12951–12958 (2020). [Online]. <https://doi.org/10.1609/AAAI.V34I07.6994>
37. Chen, F., et al.: JSPNet: learning joint semantic instance segmentation of point clouds via feature self-similarity and cross-task probability. *Pattern Recogn.* 122, 108250 (2022). <https://doi.org/10.1016/j.patcog.2021.108250>
38. He, T., et al.: Learning and memorizing representative prototypes for 3D point cloud semantic and instance segmentation. In: Vedaldi, A., et al. (eds.) *Computer Vision – ECCV 2020*. Springer International Publishing (2020)
39. Yi, L., et al.: SyncSpecCNN: synchronized spectral CNN for 3D shape segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6584–6592 (2017)
40. Klokov, R., Lempitsky, V.: Escape from cells: deep Kd-networks for the recognition of 3D point cloud models. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 863–872 (2017)
41. Wang, Y., et al.: Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* 38(5), 1–12 (2018). <https://doi.org/10.1145/3326362>
42. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567 (2018)
43. Tran, D.V., Navarin, N., Sperduti, A.: On filter size in graph convolutional networks. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1534–1541 (2018)
44. Lu, Q., et al.: PointNGCNN: deep convolutional networks on 3D point clouds with neighborhood graph filters. *Comput. Graph.* 86(Feb), 42–51 (2020). <https://doi.org/10.1016/j.cag.2019.11.005>
45. Chen, C., et al.: ClusterNet: deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4989–4997 (2019)
46. Lei, H., Akhtar, N., Mian, A.: SegGCN: efficient 3D point cloud segmentation with fuzzy spherical kernel. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11608–11617 (2020)
47. Guo, R., et al.: Point cloud classification by dynamic graph CNN with adaptive feature fusion. *IET Comput. Vis.* 15(3), 235–244 (2021). [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cvi.2.12039>
48. Wu, Z., et al.: Representing long-range context for graph neural networks with global attention. *Adv. Neural Inf. Process. Syst.* 34, 13266–13279 (2021). [Online]. Available: <https://proceedings.neurips.cc/paper/2021/file/6e67691b60ed3e4a55935261314dd534-Paper.pdf>
49. Srinivas, A., et al.: Bottleneck transformers for visual recognition. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16514–16524 (2021)
50. Zhang, Z., et al.: Depth-based subgraph convolutional auto-encoder for network representation learning. *Pattern Recogn.* 90, 363–376 (2019). <https://doi.org/10.1016/j.patcog.2019.01.045>
51. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9224–9232 (2018)
52. Riegler, G., Ulusoy, A.O., Geiger, A.: OctNet: learning deep 3D representations at high resolutions. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3577–3586 (2017)
53. Meng, H.-Y., et al.: VV-Net: voxel vae net with group convolutions for point cloud segmentation. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8499–8507 (2019)
54. Le, T., Duan, Y.: PointGrid: a deep network for 3D shape understanding. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9204–9214 (2018)
55. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29–38 (2017)
56. Xun Xu, G.H.L.: Weakly supervised semantic point cloud segmentation: towards 10x fewer labels. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13703–13712 (2020)
57. Xu, Q., et al.: Grid-GCN for fast and scalable point cloud learning. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5660–5669 (2020)

58. Yi, L., et al.: A scalable active framework for region annotation in 3D shape collections. *ACM Trans. Graph.* 35(6), 210:1–210:12 (2016). [Online]. <https://doi.org/10.1145/2980179.2980238>
59. Armeni, I., et al.: 3D semantic parsing of large-scale indoor spaces. In: *Computer Vision and Pattern Recognition*, pp. 1534–1543 (2016)
60. Lin, H., et al.: ScanNet: a fast and dense scanning framework for metastatic breast cancer detection from whole-slide image. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 539–546 (2018)
61. Li, R., et al.: Adaptive graph convolutional neural networks. In: *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)* (2018). [Online]. Available: <http://arxiv.org/abs/1801.03226>
62. Wijaya, K.T., Paek, D.-H., Kong, S.-H.: Advanced feature learning on point clouds using multi-resolution features and learnable pooling. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
63. Topping, J., et al.: Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv e-prints* (2021)
64. Wang, P., et al.: O-CNN: octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph.* 36(4), 72 (2017). <https://doi.org/10.1145/3072959.3073608>
65. Shen, Y., et al.: Mining point cloud local structures by kernel correlation and graph pooling. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4548–4557 (2018)
66. Wu, W., et al.: SK-Net: deep learning on point cloud via end-to-end discovery of spatial keypoints. *Proc. AAAI Conf. Artif. Intell.* 34, 6422–6429 (2020). <https://doi.org/10.1609/aaai.v34i04.6113>
67. Duan, Y., et al.: Structural relational reasoning of point clouds. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 949–958 (2019)
68. GeoffreyCanright, K.-M., Engo-Monsen, K.: Roles in networks. *Sci. Comput. Program.* 53(2), 195–214 (2004). *Topics in System Administration*. <https://doi.org/10.1016/j.scico.2003.12.008>
69. Zhang, Z., Hua, B.-S., Yeung, S.-K.: ShellNet: efficient point cloud convolutional neural networks using concentric shells statistics. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1607–1616 (2019)
70. Qiu, S., Anwar, S., Barnes, N.: Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1757–1767 (2021)
71. Dai, A., et al.: ScanNet: richly-annotated 3D reconstructions of indoor scenes. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2443 (2017)

How to cite this article: Zhang, W.J., et al.: Long short-distance topology modelling of 3D point cloud segmentation with a graph convolution neural network. *IET Comput. Vis.* 1–14 (2022). <https://doi.org/10.1049/cvi.2.12160>