Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049450

ARTICLE

SGT-Net: A Transformer-Based Stratified Graph Convolutional Network for 3D Point Cloud Semantic Segmentation

Suyi Liu[1,*], Jianning Chi[1], Chengdong Wu[1], Fang Xu[2,3,4] and Xiaosheng Yu[1]

[1]Faculty of Robot Science and Engineering, Northeastern University, Shenyang, 110167, China

[2]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China

[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, 110169, China

[4]SIASUN Robot & Automation Co., Ltd., Shenyang, 110169, China

*Corresponding Author: Suyi Liu. Email: 2010694@stu.neu.edu.cn

Received: 08 January 2024 Accepted: 15 April 2024 Published: 20 June 2024

ABSTRACT

In recent years, semantic segmentation on 3D point cloud data has attracted much attention. Unlike 2D images where pixels distribute regularly in the image domain, 3D point clouds in non-Euclidean space are irregular and inherently sparse. Therefore, it is very difficult to extract long-range contexts and effectively aggregate local features for semantic segmentation in 3D point cloud space. Most current methods either focus on local feature aggregation or long-range context dependency, but fail to directly establish a global-local feature extractor to complete the point cloud semantic segmentation tasks. In this paper, we propose a Transformer-based stratified graph convolutional network (SGT-Net), which enlarges the effective receptive field and builds direct long-range dependency. Specifically, we first propose a novel dense-sparse sampling strategy that provides dense local vertices and sparse long-distance vertices for subsequent graph convolutional network (GCN). Secondly, we propose a multi-key self-attention mechanism based on the Transformer to further weight augmentation for crucial neighboring relationships and enlarge the effective receptive field. In addition, to further improve the efficiency of the network, we propose a similarity measurement module to determine whether the neighborhood near the center point is effective. We demonstrate the validity and superiority of our method on the S3DIS and ShapeNet datasets. Through ablation experiments and segmentation visualization, we verify that the SGT model can improve the performance of the point cloud semantic segmentation.

KEYWORDS

3D point cloud; semantic segmentation; long-range contexts; global-local feature; graph convolutional network; dense-sparse sampling strategy

1 Introduction

Semantic segmentation on point cloud becomes a research hotspot in 3D vision, which is applied to various applications such as virtual reality [1], robot visual grabbing [2], and automatic driving [3]. Point cloud data with the advantages of depth information, able to accurately capture the spatial features. However, unlike 2D images, 3D point cloud data have congenital disadvantages such as irregular arrangement, uneven density, and sparsity in continuous space. They are usually represented by voxel, mesh, or point-based features. But voxel, mesh representations commonly have insufficient resolution, high memory cost, and are not directly related to 3D sensor output. Therefore, it is necessary to explore an advanced method to directly process point cloud data.

Previously, 3D semantic segmentation [4–8] based on deep learning makes abundant achievements, which can be classified into voxel based methods [9,10] and point based methods [4,5]. The voxel based methods project point clouds onto regular grids and convert them into voxels. Subsequently, the data is processed using variants of convolutional neural networks (CNN). However, the large amount of data preprocessing and high memory footprint limit the development of voxel-based methods. To process irregular, sparse, and unstructured data, like 3D point clouds, most studies shift the focus of point cloud processing to point-based methods. PointNet [4] is a pioneering deep learning framework for point cloud processing. Due to its global max pooling operation, local features are ignored. To better consider local information, some researchers have achieved promising results in many tasks such as image recognition [11–13] and semantic segmentation by aggregating local features. However, most of them utilize convolution operations to aggregate local features of point clouds, but ignore the establishment of long-range dependencies.

Along other lines of research, GCN utilizes points as vertices and the relationship between points as edges to construct graphs representing non-Euclidean data. Furthermore, GCN can be applied to enhance connections between nodes within features. However, only limited attempts [14–18] apply GCN to 3D point clouds. Motivated by the above works, we develop an efficient feature extractor based on GCN and K Nearest Neighbors (KNN) to capture long-range contexts and neighborhood information. Firstly, the 3D space is parted into non-overlapping cubic windows. Then a dense-sparse sampling strategy is proposed, instead, each vertex only selects the points in the neighborhood after the KNN search, we also sample the distant points as vertices. In this way, each vertex has both neighborhood points and long-range points, which effectively establish long-range context dependency and achieve a significantly enlarged receptive field. In addition, to select whether the neighborhood points searched by KNN around the current center vertex are valid, we propose a similarity measurement module to filter out noise points that may belong to different classes. This can not only improve the precision of network segmentation but also greatly improve the efficiency of the network.

On the other hand, the application of Transformer [19] in point cloud [8,20–22] has received more and more attention in recent years, which can harvest long-range context information by self-attention mechanism. "Vector self-attention" and "subtraction relation" are proposed by Point Transformer [20] for classification and dense prediction of point clouds. Offset-attention with normalization refinement and implicit Laplace operator is proposed by Point Cloud Transformer [8] to aggregate local features. SGT-GCN [23] utilizes a GCN and self-attention to enhance semantic representations by aggregating neighborhood information and focusing on vital relationships. Inspired by the above works, further to emphasize meaningful relationships among the center points, neighborhood points, and distant points, we propose a stratified self-attention mechanism based on Transformer. We utilize the dense-sparse sampling strategy mentioned above to make each "query" have both a dense "Key" at a close distance and a sparse "Key" at a long distance. In this way, the proposed self-attention mechanism redistributes the weight of the relationship between features to further enhance the most vital connections.

The advantages of our proposed SGT module are verified by extensive experiments. However, it is notable that the irregular point distribution and density diversity in the point cloud bring great challenges to the design of 3D GCN. Inspired by 2D CNN Maxpooling operation, graph maximum pool operation is utilized in the SGT module to deal with point cloud features at different scales. As a result, our method can efficiently extract the structure information of irregular 3D point clouds with any shape and size. Moreover, we aim to deem each vertex as a 3D kernel whose shape and weight can be learned during the training phase, which is conducive to faster convergence and stronger performance. The main contributions of our works can be summarized as follows:

1. We propose a novel Transformer-based stratified graph convolutional network for semantic segmentation on the point cloud, enlarging the effective receptive field and building direct long-range dependency.
2. The dense-sparse sampling strategy with similarity measurement is proposed to ensure that the neighbor points searched by KNN are similar to the central points and improve the network efficiency.
3. To further identify the most important connections, we develop a multi-key self-attention mechanism to redistribute the weight of the relationship among the center point, neighborhood points, and distant points.

2 Related Work

2.1 Point Based Semantic Segmentation Methods

Due to the irregularity of the point cloud, it was difficult to describe its spatial shape. Therefore, it was impossible to perform the convolution operation on the point cloud like 2D CNN to extract its features. Some previous works proposed various point-based methods to learn high- dimensional semantic features. PointNet [4] and its variant PointNet++ [5] solved the disorder and permutation invariance of point clouds by symmetric function (Maxpooling) and multi-layer perceptron (MLP), which became the first point-based deep learning method for point cloud analysis. Although many methods [6,17] outperformed PointNet and pointNet++ in terms of performance, most networks were based on this architecture. A positional adaptive convolution (PAConv) [24] was proposed with dynamic kernel assembly, whose convolution kernel was assembled from multiple elementary weight matrices, and the weight matrix coefficients were obtained by adaptive learning to better handle irregular and disordered point clouds. Thomas et al. [16] tried to use discrete kernel points to mimic a continuous convolution kernel. Both of the above studies used 3D convolutional kernels to extract features from point clouds, different from both, SGT took the hierarchical graph convolutional network as the main model. References [25–27] decoded the encoded point cloud into two parallel semantic and instance segmentation channels, which jointed semantic and instance features to improve the segmentation performance of the two tasks. RandLA-Net [28] was proposed using a random sampling strategy, which was suitable for large-scale outdoor point cloud processing. But this will lose key point information, resulting in low accuracy of boundary segmentation. In this paper, we proposed a dense-sparse sampling strategy with similarity measurement to reduce the negative impact of random sampling at a low computational cost. DCNet [29] built a novel feature aggregation method to relieve the key feature loss issue. However, most methods focused on aggregating local features or developing global features, failed to directly capture long-range context information and enlarge the effective receptive field. It has been demonstrated to be effective in capturing contexts from a long distance. In addition, some methods required more complex 3D convolution operations, resulting in a large amount of memory and computation, which was not available on mobile devices.

2.2 *Graph Convolutional Networks*

Graph convolution network (GCN) was a deep learning model based on graph structure, which was mainly applicable to unstructured non-Euclidean data [30]. GCN could learn and aggregate node characteristics, and weighted aggregation was used to complete the prediction task. Point cloud, as a representative of non-Euclidean data, its structure was very suitable for the GCN model. Recently, GCN became more and more popular in 2D image understanding [31], which utilized convolution neural network commonly used in images to solve the problem of non-Euclidean data. Due to the continuous growth of the computing power of graphics processing unit (GPU), GCN was applied to the field of 3D vision. DGCNN [17] utilized a new neural network module EdgeConv to deal with the classification and segmentation of 3D point clouds, which became the first method to apply GCN to point clouds. Because DGCNN is large, a linked dynamic graph CNN (LDGCNN) [32] was proposed to remove the transformation network. LDGCNN applied KNN and shared MLP to extract local features in the central point and its neighbors. Shortcuts were added between the different layers to link the hierarchical features to calculate useful edge vectors. VA-DGCNN [33] proposed a novel, feature-preserving vicinity abstraction (VA) layer for the EdgeConv module. Unlike the original DGCNN, local information is aggregated before further processing, rather than processed one point at a time with neighbors. These methods are improvements on the original DGCNN to achieve better results. But unlike us, we utilize GCN and multi-key self-attention to enhance semantic representation by aggregating neighborhood information and focusing on important relationships. GAPointNet [34] proposed a novel neural network for point cloud analysis, which was able to learn local geometric representations by embedding graph attention mechanism within stacked MLP layers. Lin et al. [18] proposed a 3D graph convolution network, which learned 3D kernels with graph max-pooling mechanisms for extracting geometric features from point cloud data across different scales. Kim et al. [35] developed a low-power graph convolutional network for mobile devices, which greatly reduced memory and computation. Our model and the above methods [18,35] are both lightweight segmentation networks based on GCN. However, we did not use random sampling to reduce the size of the model. The above attempts only used the traditional KNN search algorithm to form local neighborhoods to aggregate local features, which was still difficult to build long-range dependency. To address this, Song et al. [36] proposed a global affinity adaptation module to adapt global priors to the sample via a graph convolutional network built over different categories. Different from the above works, we developed an efficient feature extractor based on GCN and KNN to enlarge the effective receptive field and building direct long-range dependency.

2.3 *Transformer for Vision*

The Transformer network was applied to the field of computer vision originally by Vision in Transformer (VIT) [37], which used patch embedding and Transformer to extract encoder features for image classification. However, the features extracted by VIT were relatively rough and could not complete the detailed tasks. Subsequently, there were several other methods based on Transformer, such as object detection [38], image recognition [13], and image super resolution [39] for 2D image analysis. Most of them applied local and global attention, through convolution, MLP, or linear layer to establish long-range dependencies.

Due to the brilliant success of Transformer in the field of 2D images, which were attracted more attention in 3D point cloud. Point Transformer [20] successfully introduced Transformer into the field of point cloud processing. Guo et al. [8] developed a neighbor embedding mechanism achieved by EdgeConv. A BERT-style pre-training strategy for 3D global Transformer [21] was proposed, which generalized the concept of BERT to 3D point cloud processing. Unlike the above methods, we proposed a multi-key self-attention mechanism to obtain two different scales of attention maps, which achieved a significantly enlarged effective receptive field.

3 Method

In this section, we introduce the framework of SGT shown in Fig. 1, which contains stratified sample module, graph convolutional module, and Transformer block. The SGT captures long-range context information and enlarges the effective receptive field. In addition, the weight of the relationship among the center point, neighborhood points, and distant points is redistributed by self-attention mechanism. It aims to solve the semantic segmentation problem of complex point cloud environment and reduce the memory occupation of the network.

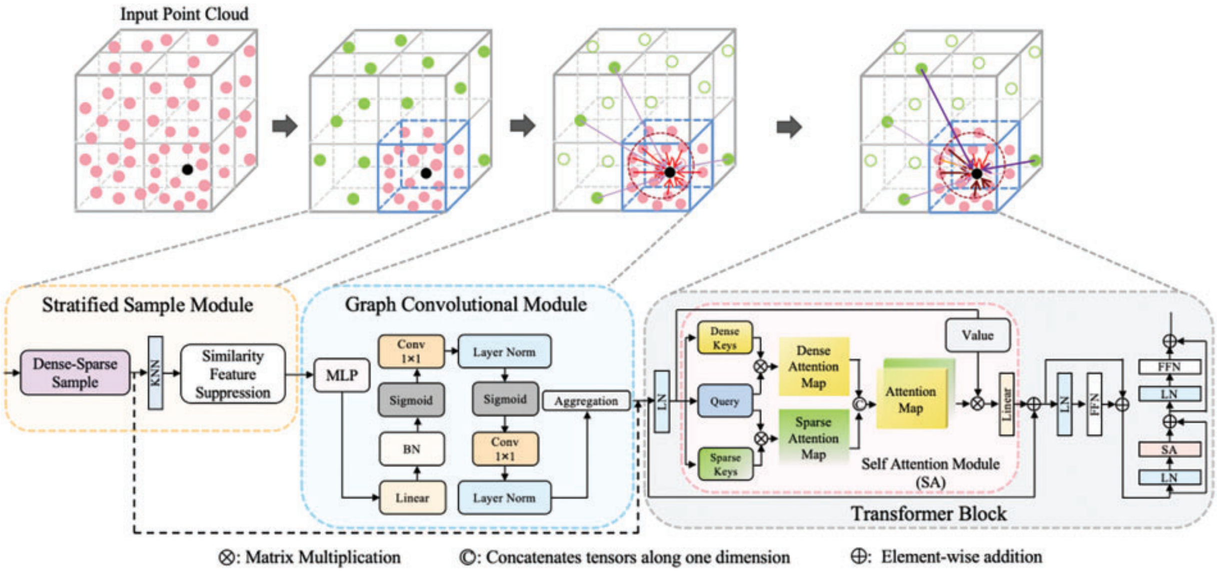

Figure 1: The framework of the proposed 3D semantic segmentation method. The center point (black), dense points in close range (pink), and sparse points in long range (green). The solid points in the third cube are sampling points, while the hollow points are not. The distant edges are established by the GCN module (pink lines), and the near edges (red lines). After the self-attention module, dark connections indicate a large weight, and light connections indicate a small weight

For the convenience of the following description, we define the several notations as follows. The raw point cloud sampled by the farthest point can be regarded as a sequence, which contains a total of n points $N = \{N_i \in \mathbb{R}, i = 1, 2, \cdots, n\}$ located on the surface of the point cloud instance. $f(N_i) \in \mathbb{R}^F$ represents the features of each point, where F represents the dimension of the point features. Typical features contain coordinates $\left(x_{N_i}, y_{N_i}, z_{N_i}\right)$, normal vectors $\left(v^x_{N_i}, v^y_{N_i}, v^z_{N_i}\right)$, and RGB color information $\left(r_{N_i}, g_{N_i}, b_{N_i}\right)$. In this study, we set $F = 9$, and only use typical features mentioned above as the input features.

3.1 Stratified Sample Module

Current semantic segmentation methods mostly only use the traditional KNN search algorithm to generate local neighborhoods to aggregate local features. It is still difficult to interact the information with long-range context. Therefore, to solve the above problems by only adding negligible additional computation, we propose a dense sparse sampling strategy that aggregates local features and extends connectivity. The input point clouds are divided into overlapping cube windows with different scales, as shown in Fig. 2.

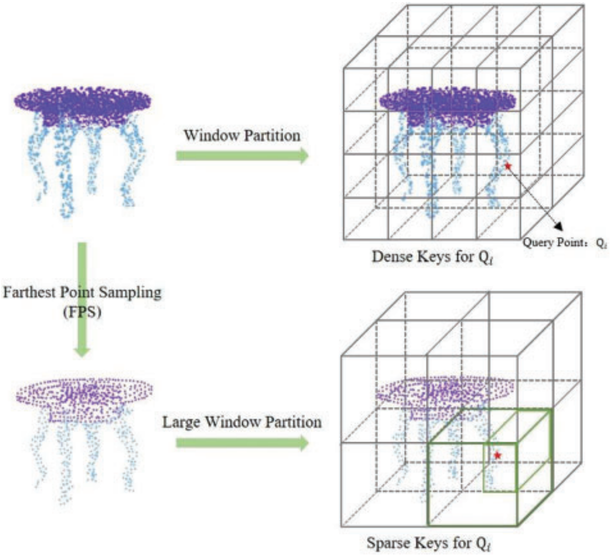

Figure 2: Illustration of the dense-sparse strategy for keys sampling

The lower right side of Fig. 2 shows small windows with dense points, and large windows with sparse points in other parts. The farthest point sampling (FPS) is used on the scale of s to reduce the input dense point cloud to sparse point cloud. Following this process, we utilize KNN to search the center points as the vertices. As shown in Fig. 3, a center vertex not only has neighbor points searched by KNN, we randomly select k points from long-range sparse points. In this way, a central vertex has $j+k$ neighbor points, which can expand the receptive field for subsequent graph convolutional module.

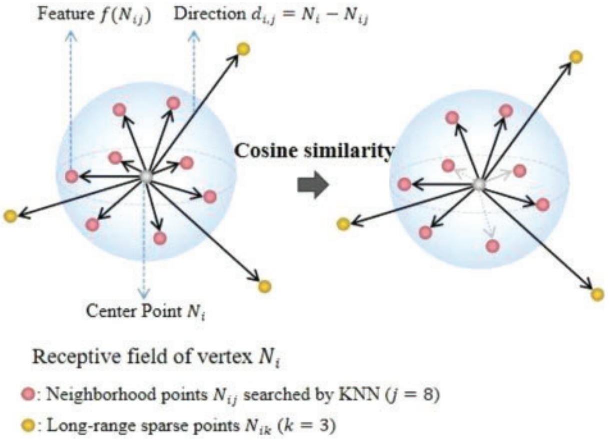
Feature $f(N_{ij})$
Direction $d_{i,j} = N_i - N_{ij}$
Cosine similarity
Center Point N_i
Receptive field of vertex N_i
●: Neighborhood points N_{ij} searched by KNN ($j = 8$)
●: Long-range sparse points N_{ik} ($k = 3$)

Figure 3: Illustration of receptive field of vertex N_i

In order to ensure that the neighbor points searched by KNN are similar to the central vertex and improve the network efficiency, we add similarity measurement in the stratified sample module.

When instances of different semantic categories are close to each other, the accuracy of instance edge segmentation will be reduced.

The main reason is that GCN uses KNN to search for neighbor points, that is, points with a relatively close distance, which results to classify them into one class. Therefore, we use cosine similarity instead of distance similarity to filter neighborhood points. Two instances that come together more closely are not necessarily of the same semantic class. It could be two completely different semantic categories, or it could be a noise point around a central point. Therefore, distance alone cannot accurately judge the similarity between points, but we can utilize the color information of the point cloud to judge the similarity between the search point and the center by cosine similarity. On the contrary, if two points are relatively far apart in space, using only distance as a measure of similarity is likely to be judged as uncorrelated. However, in indoor scenes, there may be instances with larger volumes, and even if two points are far apart, they may still belong to the same semantic category. Therefore, we use the cosine similarity measure to help our network improve segmentation accuracy and reduce the memory and computation of the network. Cosine similarity is defined as Eq. (1):

$$d\left(N_i, N_{ij}\right) = \frac{f\left(N_i\right)^T \cdot f\left(N_{ij}\right)}{\left\|f\left(N_i\right)\right\| \cdot \left\|f\left(N_{ij}\right)\right\|} \tag{1}$$

where $\|\cdot\|$ represents the vector module.

3.2 Graph Convolutional Module

The point cloud is non-Euclidean data, so it is difficult to process it with traditional convolution neural network. In addition, for point cloud analysis, we need to consider both the feature information and structure information of the points. If feature extraction is done manually, many hidden and complex patterns will be lost and complicated calculations will be brought out. Therefore, the graph convolution neural network is utilized to analyze the point cloud structure. Because GCN needs to transfer the point cloud structure as a graph structure, but the input point cloud after down-sampling is still large, we need to use the KNN search method described in the previous section to perform the graph convolution operation on the local part of the point cloud. In this process, a graph G is defined as a tuple $G = (V, E)$, where $V = \{v_i \,|v_i \in V\}$ is a set of vertices, and $E = \left\{e_{i,j} = v_i - v_j \,\middle|\, v_i, v_j \in V\right\}$ represents the connectivity between vertices. Specifically, $v_i = f\left(N_i\right)$ is the feature of the center vertex in the ith cell, which contains position features and other information. The set of k-nearest neighbors for the vertex v_i is denoted R_i and the edge of the graph is defined as E. It is worth emphasizing that when we apply it to neural networks, the directed graph structure ignores the order of vertices, and each vertex is propagated.

An asymmetric function combining neighboring information and shape structure is introduced by DGCNN to establish topological relationships between vertices. This asymmetric function is adopted in our study and defined as Eq. (2):

$$f\left(e_{ij}\right) = RELU\left(\theta_m \cdot \left(N_{ij} - N_i\right) + \phi_m \cdot N_i\right) \tag{2}$$

3.3 Transformer Block

GCN is used to extract geometric relationships, but the extracted geometric relationships are not necessarily key relationships. Therefore, we need to introduce a Transformer block to redistribute the weight of the relationship among the center point, neighborhood points, and distant points. The Transformer block is composed of a multi-key self-attention module and a feed-forward network (FFN). The encoder and decoder in Transformer block both use the multi-key self-attention module.

Since every query point only attends to the local points in its own window, the vanilla version Transformer block suffers from limited effective receptive field even with a shifted window. Therefore, it fails to capture long-range contextual dependencies over distant objects, causing false predictions. To adequately reflect long distance and neighborhood points dependencies, in this section, we use the dense-sparse Keys strategy shown in Fig. 2. For each query point Q_i there are two keys, i.e., K_i^{dense} and K_i^{sparse}. K_i^{dense} represents the dense Key matrixes sampled from the small cubes in Fig. 2, and K_i^{sparse} represents the sparse Key matrixes sampled from the large cubes. Following common practice, we use the original Transformer to get dense and sparse attention maps. Finally, the attention maps are obtained through feature concatenation. Due to the hierarchical strategy of key sampling, the effective receiving field is significantly expanded, and the query features can effectively aggregate long-range context. Compared to the regular version, we only generate negligible additional calculations on sparse remote keys. We set the number of points in the k-th cube to be P_k, and given that N_h is the number of heads, the dimension of each head is N_d. Therefore, $N_a = N_h \times N_c$ is the feature dimension. For the set of input points in k-th cube, $N_k = \left\{n_k^1, n_k^2 n_k^3, \cdots, n_k^n\right\} \in R^{P_k \times N_h \times N_d}$, the multi-head self-attention in the k-th cube is formulated as follows:

$$K^{dense} = LN_{K^{dense}}(N), K^{sparse} = LN_{K^{sparse}}(N) \tag{3}$$

$$Q = LN_Q(N), V = LN_V(N) \tag{4}$$

$$Attn^{dense} = Q \times K^{dense}, Attn^{sparse} = Q \times K^{sparse} \tag{5}$$

$$Attn = softmax\left(Attn^{dense} \cup Attn^{sparse}\right) \tag{6}$$

$$Y = \sum Attn \times V, Z = LN(Y) \tag{7}$$

where $Q, K, V \in R^{P_k \times N_h \times N_d}$ are obtained from N_k by three shared linear layers $LN(\cdot)$. $Attn \in R^{P_k \times P_k \times N_h}$ is concatenated from local feature maps as well as sparse feature maps that can establish long distance dependencies. $Y, Z \in R^{P_k \times N_h \times N_d}$ are the aggregated features and output features, respectively. In addition, LayerNorm before each self-attention module or feed-forward network is used.

3.4 Semantic Segmentation Network

In the 2D world, semantic segmentation tasks have received increasing attention and have achieved great achievements. Previous works [40–42] have proposed various feature aggregation methods and cross layer connections. Motivated by them, we propose a U-shaped model structure, illustrated in Fig. 4. The backbone of this architecture is consisted of several SGT modules mentioned above, point feature embedding module, down sampling modules, and up sampling modules. Where N represents the number of point clouds, r represents the down sampling rate, and D represents the feature dimension of the points.

Because down sampling is necessary to improve network processing speed and reduce memory consumption, we adopt the farthest point sampling in [20] to identify a subset with the requisite cardinality. Then, each input feature goes through a linear layer, a batch normalization layer (BN), an activation function, and Maxpooling to complete the down sampling operation.

For up sampling, we also use the method in [20]. Each input point feature is processed by a linear layer, followed by a BN layer and an activation function, then the subset of the input point set is mapped onto the higher-resolution point set by trilinear interpolation.

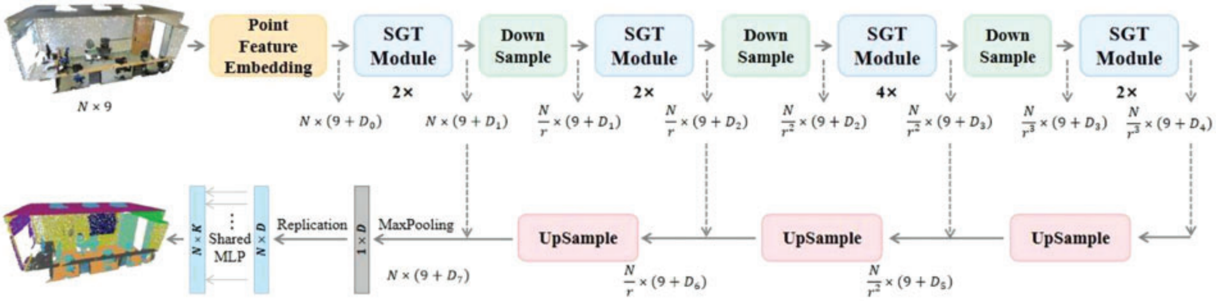

Figure 4: Architecture of the semantic segmentation network

To increase the local dependency of point cloud learning, the first layer of our network uses point embedding to establish local geometric context information. Most current methods use linear layers or MLP to map XYZ position information and RGB color information to a high-dimensional feature space. However, the simple use of a linear layer network has a slow convergence rate and also increases the network parameters, which affects the network performance. The point feature from MLP only contains its own xyz and RGB information and lacks local relevance. Therefore, we use the strategy of KPConv [16] for local aggregation and only generate negligible additional computation.

4 Experiments

4.1 Datasets

To verify the performance of the proposed point cloud segmentation algorithm, we utilize two public available datasets for experimental validation, namely S3DIS [43] dataset and ShapeNet [44]. The S3DIS dataset is a large-scale real scene 3D segmentation dataset that consists of six areas scanned by a scanner called Matterport, including 272 rooms and approximately 215 million points. Each point in the scene point cloud contains an instance label and a semantic label; In addition to the large-scale real-world scene benchmark S3DIS, we also evaluated our approach on the ShapeNet part dataset. ShapeNet is a constructed synthetic virtual segmentation dataset containing 16 classes with 16881 3D objects. Each point sampled from the shapes is assigned with one of the 50 different parts. The instance annotations from [42] are used as the instance ground-truth labels.

4.2 Network Configuration

The model architecture is shown in Fig. 4. The feature extracting part is composed of point feature embedding module, down sample module, and SGT module. In the SGT module, we set neighbor number $N_{ij} = 30$, and long-range sparse points $N_{ik} = 10$ for the receptive field. In addition, during the encoding stage, four stages are constructed with the SGT module depths [2,2,4,2]. Each time the SGT module is passed, the network will down sample the number of point clouds, but the characteristic dimension of the points increases. In the decoder architecture, we upsample and concatenate large size features, which is formulated as [20]. The decoded features obtained through three up sampling operations and feature concatenating are sent to the shared MLP layers after Maxpooling and replication operations, and the final semantic prediction $N \times K$ is output.

4.3 Implementation Details

For the S3DIS dataset, each point has 9 dimensions of features, which are the location information of the point (X, Y, Z), the color features (R, G, B), and the normalized coordinates of the indoor scene.

We set the feature dimensions after point embedding layer to 64, and the number of self-attentive heads in trans is set to 3. Each down sampling is 1/4 of the number of points in the previous layer, but the number of the point features and self-attention heads is doubled. Following common practice, we adopt the strategy proposed by [4] to slice the input point cloud into small cubes, each cube containing 4096 points. We use 4 Titan GPUs to train our network for 600 epochs with an Adam optimizer. The batch size is set to 8, the initial learning rate is set to 0.001, and the learning rate is halved for every 20,000 iterations. For the ShapeNet dataset, each object consists of 2048 points, and each point has only 3 dimensional position information (X, Y, Z). We calculate the overall accuracy (oAcc), mean accuracy (mAcc), intersection over union (IOU) for each category, and mean intersection over union (mIOU) with a threshold of 0.5 as evaluation metrics for the semantic segmentation task.

4.4 Results

4.4.1 S3DIS Dataset

For the S3DIS dataset, we explicitly evaluated the performance of our method on area 5 and the results of the 6-fold cross validation. Note that it is a common practice to analyze the performance of 6-fold cross validation separately, but area 5 on the S3DIS dataset is not in the training set. Therefore, evaluating area 5 can well test the generalization performance of the network. We compare the classical semantic segmentation methods [5,8,16,28,45], the semantic and instance joint segmentation methods [25,26], the GCN based semantic segmentation methods [17,18,29,33], and the results are shown in Table 1.

Table 1: Comparison per-class performance of our proposed method with state-of-the-art on S3DIS area 5 dataset

Method	mIou	mAcc	oAcc	cei.	flo.	wall	bea.	col.	win.	door	table	cha.	sofa	boo.	boa.	clu.
PointNet++ [5]	51.5	–	83.8	91.4	97.9	74.3	0.0	3.7	48.9	36.3	69.4	76.2	26.5	53.5	49.3	41.9
PCT [8]	61.3	67.7	–	92.5	98.4	80.6	0.0	19.4	61.4	48.0	76.6	85.2	46.2	67.7	**67.9**	52.3
KPConv [16]	**67.1**	72.8	–	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	**91.0**	**75.4**	75.3	66.7	**58.9**
DGCNN [17]	49.0	–	83.2	91.1	97.3	74.5	0.0	11.9	49.5	33.5	66.9	69.4	20.5	47.5	34.7	40.8
3D-GCN [18]	51.9	–	84.6	91.4	97.1	75.9	0.1	22.3	43.5	30.1	71.5	79.4	21.9	53.7	42.9	44.9
ASIS [25]	53.4	60.9	86.9	92.0	98.0	75.3	0.0	10.1	49.9	24.2	72.9	78.1	33.4	58.4	51.0	50.7
JSNet++ [26]	58.0	–	–	**93.7**	**98.5**	80.5	0.0	16.9	57.2	41.9	76.8	84.7	30.5	60.2	58.3	54.9
RandLA-Net [28]	62.4	71.4	87.2	91.1	95.6	80.2	0.0	24.7	**62.3**	47.7	76.2	83.7	60.2	71.1	65.7	53.8
Ours	66.4	**73.4**	**87.9**	92.5	97.6	**83.9**	0.0	**27.6**	61.0	**72.1**	**84.4**	86.1	63.1	**76.9**	64.5	54.4

We notice that all methods perform similarly in the ceiling, floor, and beam categories, as they can be easily classified based on the location in the room. However, our method achieved better results in the other categories indicating that the SGT module can capture long-range context information and enlarge the effective receptive field. This is important for background segmentation in indoor scenes. In addition, the segmentation accuracy of our method for foreground categories such as door, table, chair, and clutter is better than the comparison methods. It indicates that our method not only captures long-distance information but also has a stronger ability to recognize local geometric shapes. With the benefit of the complementarity between GCN and Transformer, the SAT-Net finally achieves a leading performance of 73.4 in mAcc and 87.9 in oAcc, respectively.

To avoid overfitting on S3DIS area 5, we further verify the generalization performance of our method by using the 6-fold cross validation, nd the results are shown in Table 2. The performance of SGT-Net is relatively balanced across various categories, achieving the best performance in six of them. Specifically, the proposed SGT-Net can not only achieve leading performance in long-range background category segmentation tasks, such as wall, beam and column, but also achieve optimal performance in short-range foreground category segmentation tasks, such as table, bookcase and clutter.

Table 2: Comparison per-class performance of our proposed method with state-of-the-art on S3DIS with 6-fold cross validation

Method	mIou	mAcc	oAcc	cei.	flo.	wall	bea.	col.	win.	door	tab.	cha.	sofa	boo.	boa.	clu.
PointNet++ [5]	57.6	–	83.5	91.7	93.9	73.5	54.7	20.7	53.0	57.0	63.0	59.3	36.4	49.0	49.2	47.0
PCT [8]	66.5	73.0	87.9	–	–	–	–	–	–	–	–	–	–	–	–	–
KPConv [16]	70.6	79.1	–	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
DGCNN [17]	56.0	–	83.4	92.0	94.5	73.9	50.4	32.5	54.9	59.2	62.5	53.9	16.6	45.7	46.3	45.4
3D-GCN [18]	60.8	–	85.8	91.7	95.5	77.2	53.0	38.4	52.3	59.0	67.6	70.8	28.1	51.5	51.9	53.2
ASIS [25]	59.3	70.1	86.2	92.1	91.8	73.7	50.4	33.9	48.4	62.5	66.2	63.4	31.5	51.2	56.1	49.8
JSNet++ [26]	62.4	–	–	**94.1**	**97.3**	78.0	41.3	32.2	52.0	70.0	69.9	72.7	37.9	54.1	51.3	60.2
RandLA-Net [28]	70.0	82.0	88.0	93.1	96.1	80.6	62.4	48.0	64.4	69.4	69.4	76.4	60.0	64.2	65.9	60.1
DCNet [29]	72.4	82.1	89.3	–	–	–	–	–	–	–	–	–	–	–	–	–
VA-DGCNN [33]	–	82.2	89.2	–	–	–	–	–	–	–	–	–	–	–	–	–
BAAF [45]	72.2	**83.1**	88.9	93.3	96.8	81.6	61.9	49.5	65.4	73.3	72.0	**83.7**	67.5	64.3	**67.0**	62.4
Ours	**72.5**	82.7	**89.5**	93.8	96.5	**84.9**	**64.0**	**54.7**	63.8	74.1	**74.9**	80.2	62.7	**75.1**	65.2	**63.1**

Although our method is inferior to JSNet++ in some background categories (such as ceiling and floor) and inferior to KPConv and BAAF in some foreground categories (such as window, door, chair and sofa), the main contribution and strength of our work is different from those contrasting approaches. JSNet++ designed a mutual promotion strategy for semantic segmentation and instance segmentation and proposed a pointwise correlation module to further improve the accuracy of semantic segmentation. KPConv proposed a convolution directly applied to point clouds to better aggregate local spatial features. However, it did not consider the context of global features, and the calculation cost is large. BAAF introduced a bilateral block to augment the local context of the points and also ignored to extract long-range contexts. This will lose the accuracy of long-range background segmentation in the scene. Our proposed SGT-Net complements the local feature capture capability of GCN with the global feature extraction capability of the attention mechanism, and improves the interaction between global and local features by aggregating long-range and neighborhood information. Besides, we also propose a dense-sparse sampling strategy to enlarge the effective receptive field and develop a self-attention mechanism based on Transformer to redistribute the weight of the relationship among center point, neighborhood points, and distant points. It is worth noting that SGT-Net does not perform much worse in categories that are not as well as the above methods, but the above methods perform much worse in categories that are not as well as ours. For example, it is 22.7%, 22.5%, 24.8% and 20% higher than JSNet++ respectively in beam, column, sofa and bookcase. KPConv does not perform as well as SGT-Net in the segmentation of long-range background categories. BAAF is slightly better than SGT-Net in mAcc (+0.4), but its performance is not as good as ours in the long range of large object segmentation. It can be seen that our method trades off all categories and achieves leading performance of 72.5 in mIoU and 89.5 in oAcc.

Fig. 5 visualizes the segmentation results on area 5 of the S3DIS dataset. When the edges of different instances overlap or closely connected, our method can still accurately segment each instance, while the comparison methods may misclassify semantic categories. For the segmentation of complex edges, the precision of the proposed method is better than that of the comparison methods, and it is closer to the ground truth.

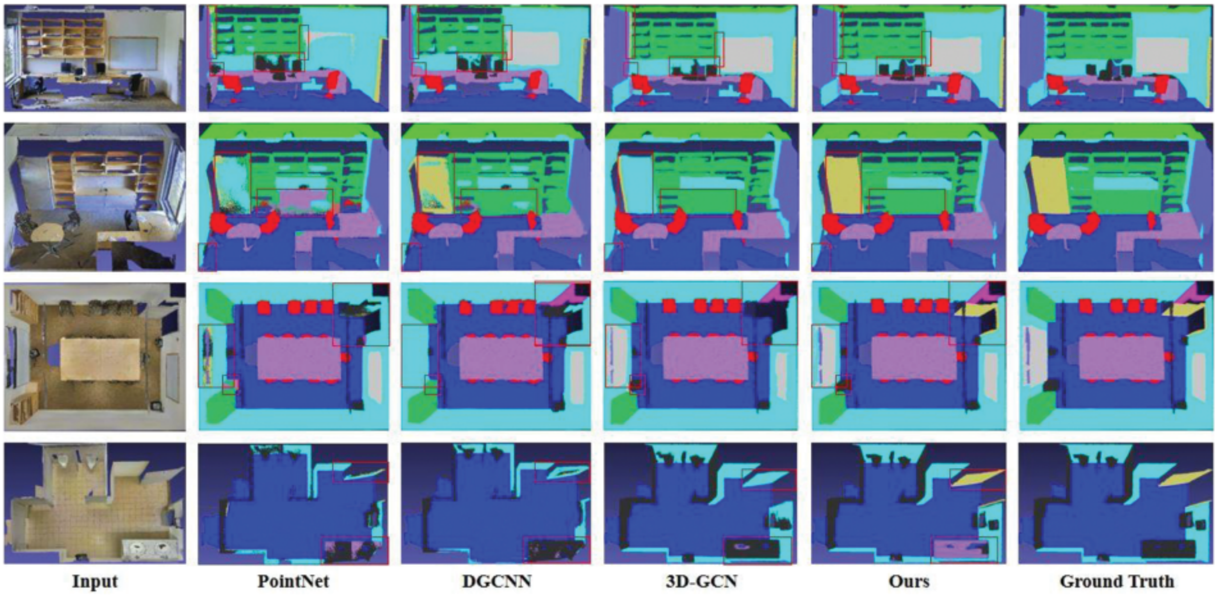

Figure 5: Comparison of our method with the state-of-the-art methods in semantic segmentation task on area 5 of S3DIS dataset. Objects of different colors represent different categories of targets. The red rectangular boxes circle the segmentation details

4.4.2 ShapeNet Dataset

Besides evaluation on the large-scale indoor dataset S3DIS, we also conduct experiments on ShapeNet [16] to evaluate the performance of our method at part level segmentation. Table 3 presents the class-wise segmentation results. We can observe that our model achieves competitive results on the ShapeNet dataset. In fact, our method and PCT (Point cloud transformer) achieved the best results in most categories. Due to the advantage of the Transformer based feature extraction method, the average accuracy of PCT is slightly better than our method. But its model parameters have also multiplied. We will discuss it in the next section. In addition, we notice that SGPN achieves the best results in certain categories as it takes the advantage of the instance segmentation tasks, which employs additional losses such as similarity matrix and confidence loss to accurately distinguish different instances with the same category. Therefore, the addition of instance information helps with its feature semantic segmentation performance. Fig. 6 visualizes the results of our method on part level segmentation tasks. Our method benefits from the similarity measurement strategy, which is more precise in segmenting part edges and closer to the ground truth.

Table 3: Comparison per-class performance of our proposed method with state-of-the-art on ShapeNet dataset

Methods	avg	air.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.	tab.
PointNet++ [5]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
PCT [8]	**86.4**	85.0	82.4	**89.0**	**81.2**	91.9	71.5	91.3	**88.1**	**86.3**	95.8	64.6	95.8	83.6	62.2	77.6	83.7
DGCNN [17]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	**63.5**	74.5	82.6
LDGCNN [32]	85.1	84.0	83.0	84.9	78.4	90.6	74.4	91.0	**88.1**	83.4	95.8	67.4	94.9	82.3	59.2	76.0	81.9
GAPointNet [34]	84.9	84.0	86.2	88.8	78.3	90.7	70.4	91.3	87.3	82.8	96.0	68.7	95.1	82.0	63.0	74.8	81.4
SGPN [46]	85.8	80.4	78.6	78.8	71.5	88.6	**78.0**	90.9	83.0	78.8	95.8	**77.8**	93.8	**87.4**	60.1	**92.3**	**89.4**
PointASNL [47]	86.1	84.1	84.7	87.9	79.7	92.2	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
Ours	86.3	**85.2**	**86.2**	87.6	80.9	**93.4**	72.8	**92.7**	87.5	83.8	**96.5**	72.3	**96.1**	84.2	63.3	73.1	83.6

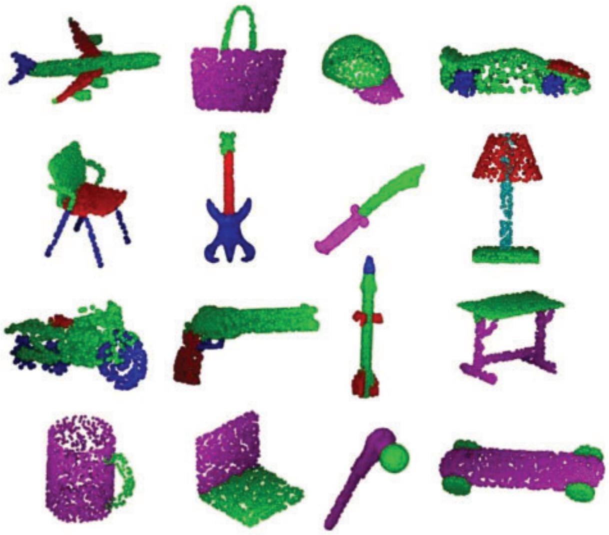

Figure 6: Part level semantic segmentation results on ShapeNet dataset. Semantic annotation using different colors for different parts

4.5 Ablation Study

4.5.1 Neighbor Number j and Long-Range Number k in the Receptive Fields

We conduct experiments on the receptive field in the stratified sampling module by varying the number of neighbor points and long-range points. We perform ablation experiments on the S3DIS dataset Area 5. The results are shown in Table 4. We can see that the insufficient or excessive number of neighbor points and long-range points can affect the performance of the network in extracting local information and establishing long distance context dependence.

When the number of neighborhood points and long-range points is small ($j = 10$, $k = 10$), the performance of segmentation is relatively poor because the network cannot extract effective features. When the number of long-range points increases, the number of noise in the sampling points will increase, which interferes with the establishment of context dependence on the network. Thus resulting in a decline in the segmentation accuracy of the network. When the number of neighborhood points and long-range points is large, the segmentation performance is close to the optimal result, but the computational cost increases. It is not difficult to find from Fig. 7 that when there are enough neighborhood points and long-range points, the robustness of the network becomes better. Therefore, we sample an appropriate number of neighboring points and long-distance points to achieve better performance.

Table 4: Impact of different numbers of neighbor points and long distance points on network performance

Neighbor number *j*	10				20				30				50			
Long-range number *k*	10	20	30	50	10	20	30	50	10	20	30	50	10	20	30	50
S3DIS area 5 (mIou)	58.6	56.1	53.9	47.7	64.5	65.1	62.7	60.5	**66.4**	65.8	64.1	61.9	64.3	66.2	65.7	64.2
ShapeNet (avg)	81.3	83.5	82.8	78.6	85.1	86.0	84.8	84.0	**86.3**	85.2	86.1	84.7	85.4	85.7	85.6	84.5

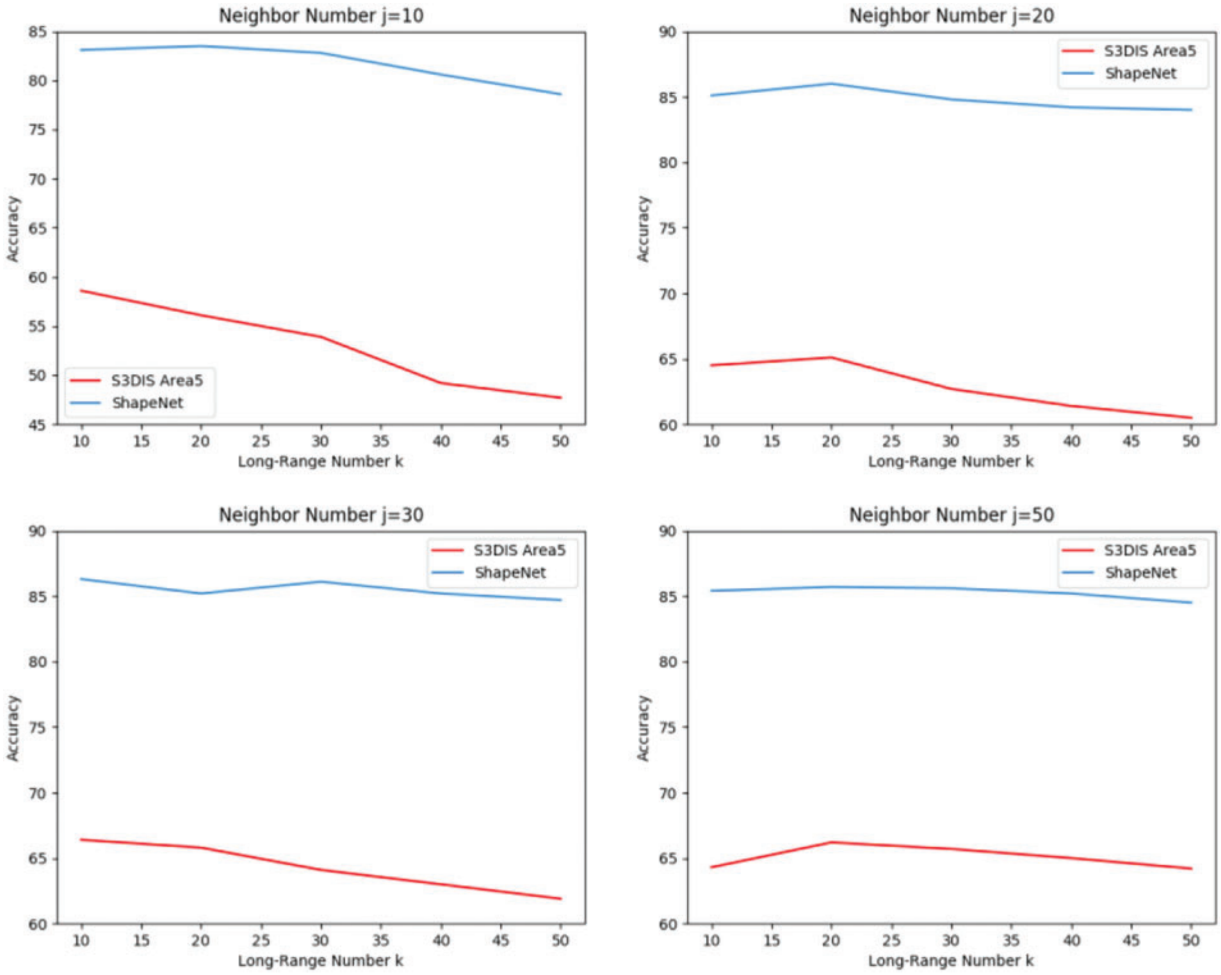

Figure 7: The influence of different number of neighbor points and long-range points on the network performance

4.5.2 Effect of Similarity Measurement Strategy and Self Attention Module

Our network includes two important modules to help the SGT module improve segmentation performance. Therefore, we conduct a series of ablation experiments to verify their effectiveness. Table 5 demonstrates the impact of each module on the backbone network. It can be seen that the best effect is achieved when two modules are working simultaneously. When the self-attention mechanism is not added to the network, but only the similarity measurement mechanism is added. We find that the segmentation accuracy is slightly better than that when neither module is working. From this, we can infer that the proposed similarity measurement mechanism can ensure that the neighbor points searched by KNN are similar to the central vertex and improve the network efficiency. On the contrary, when there is only self-attention module, the segmentation accuracy of the network is improved significantly. It proves that the self-attention module can redistribute the weight of the relationship among the center point, neighborhood points, and distant points to improve the segmentation performance. In addition, as shown in Fig. 3, the self-attention module plays an important role in long-distance background segmentation. Because Transformer is suitable for capturing global features, and our SGT module is good at local feature extraction, the two performances are just complementary.

Table 5: Effect of similarity measurement strategy and self-attention module on S3DIS area 5

Similarity	Self-attention	S3DIS area 5 (mIou)	ShapeNet (avg)
		59.1	79.0
✓		62.7	80.2
	✓	63.5	84.9
✓	✓	**66.4**	**86.3**

4.6 Complexity Analysis

We compare the number of parameters, time, and floating point operations (FLOP) between our method and the comparison methods on the ModelNet40 dataset. The overall accuracy is used to evaluate the segmentation performance. From Table 6, our model achieves segmentation performance comparable to state-of-the-art models. Although the segmentation performance is slightly lower than PCT, the number of parameters, FLOP in our model is only half of that in PCT. The number of parameters in our model is slightly higher than 3D-GCN, and the reason is that 3D-GCN uses a random sampling technique, where the subset points are sampled randomly. In addition, when the support number of the learnable kernels proposed by 3D-GCN increases, its model complexity will increase significantly, resulting in a large memory and computational load.

Table 6: Number of parameters in different models for semantic segmentation tasks on ModelNet40

Method	Params (M)	FLOPs	Time (ms)	Acc (%)
PointNet++ [5]	1.99	3136 M	32.0	90.7
PCT [8]	2.88	2327 M	–	**93.2**
DGCNN [17]	1.81	2432 M	52.0	92.9
3D-GCN [18]	**0.89**	–	**17.0**	92.1
LDGCNN [32]	1.08	–	43.0	92.9

(Continued)

Table 6 (continued)

Method	Params (M)	FLOPs	Time (ms)	Acc (%)
GAPointNet [34]	1.91	1228 M	26.0	93.0
Ours	1.16	**1029 M**	23.0	93.0

5 Conclusion

In this paper, we propose a novel stratified graph convolutional network, named SGT-Net, which can extract long-range contexts and effectively aggregate local features for semantic segmentation in 3D point cloud space. The technical contributions of our network lie in the design of the stratified graph convolution strategy and the weight allocation of attention mechanism based on the Transformer. We propose a novel Transformer-based stratified graph convolutional network for semantic segmentation on the point cloud, enlarging the effective receptive field and building direct long-range dependency. The dense-sparse sampling strategy with similarity measurement is proposed to ensure that the neighbor points searched by KNN are similar to the central points and improve the network efficiency. Experiments show that our method achieves the same or better segmentation performance than the state-of-the-art methods. Although not superior to the latest methods in some respects, we demonstrate that our model can effectively enlarge the receptive field and is computationally more efficient. In particular, SGT-Net shows a remarkable accuracy with only half the number of parameters compared to other methods, which shows great potential for real-time applications, such as robot visual grabbing. The success of this model also verifies the efficiency of graph attention networks not only in calculating the similarity of graph vertexes, but also in understanding geometric relationships.

Acknowledgement: The authors would like to thank the editors and reviewers for their valuable work, as well as the supervisor and family for their valuable support during the research process.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grant Nos. U20A20197, 62306187 and the Foundation of Ministry of Industry and Information Technology TC220H05X-04.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Suyi Liu and Jianning Chi; data collection: Suyi Liu and Chengdong Wu; analysis and interpretation of results: Suyi Liu, Jianning Chi and Xiaosheng Yu; draft manuscript preparation: Suyi Liu and Fang Xu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The training data used in this paper were obtained from S3DIS, ShapeNet, and ModelNet40. Available online via the following link: http://buildingparser.stanford.edu/dataset.html, https://www.shapenet.org/, and http://modelnet.cs.princeton.edu/.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. Wang, Y. Liu, X. Liu, and J. Wu, "Automatic virtual portals placement for efficient VR navigation," presented at the 2022 IEEE Conf. Virtual Reality 3D User Interf. Abstracts Workshops (VRW), Christchurch, New Zealand, Mar. 2022, pp. 628–629.

[2] H. Tian, W. Wu, H. Liu, Y. D. Liu, J. Zou and Y. Zhao, "Robotic grasping of pillow spring based on M-G-YOLOv5s object detection algorithm and image-based visual serving," *J. Intell. Robot. Syst. Theory Appl.*, vol. 109, no. 3, pp. 67, Nov. 2023. doi: 10.1007/s10846-023-01989-x.

[3] F. Ma, Y. Liu, S. Wang, J. Wu, W. Qi and M. Liu, "Self-supervised drivable area segmentation using LiDAR's depth information for autonomous driving," presented at the 2023 IEEE/RSJ Int. Conf. Intelligent Robots Syst. (IROS), Detroit, MI, USA, Oct. 2023, pp. 41–48.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," presented at the 30th IEEE Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, Jul. 2017, pp. 77–85.

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," presented at the 31st Int. Conf. Neural Inf. Process. Syst., Red Hook, NY, USA, 2017, pp. 5105–5114.

[6] W. Wu, Z. Qi, and F. X. Li, "PointConv: Deep convolutional networks on 3D point clouds," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 9613–9622.

[7] H. Zhao, L. Jiang, C. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 5560–5568.

[8] M. Guo, J. Cai, Z. Liu, T. Mu, and R. R. Martin, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Dec. 2020. doi: 10.1007/s41095-021-0229-5.

[9] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 4490–4499.

[10] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Boston, MA, USA, Jun. 2015, pp. 1912–1920.

[11] J. Chen, H. Ren, F. S. Chen, S. Velipasalar, and V. V. Phoha, "Gaitpoint: A gait recognition network based on point cloud analysis," presented at the IEEE Int. Conf. Image Proc. (ICIP), Bordeaux, France, Oct. 2022, pp. 1916–1920.

[12] L. Hui, M. Cheng, J. Xie, J. Yang, and M. M. Cheng, "Efficient 3D point cloud feature learning for large-scale place recognition," *IEEE Trans. Image Proc.*, vol. 31, pp. 1258–1270, Jan. 2022. doi: 10.1109/TIP.2021.3136714.

[13] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, Jun. 2020, pp. 10073–10082.

[14] J. Chen, Y. Chen, and C. Wang, "Feature graph convolution network with attentive fusion for large-scale point clouds semantic segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, no. 7, pp. 1–5, Aug. 2023. doi: 10.1109/LGRS.2023.3330882.

[15] F. Hao, J. Li, R. Song, Y. Li, and K. Cao, "Structure-aware graph convolution network for point cloud parsing," *IEEE Trans. Multimed.*, vol. 25, pp. 7025–7036, Oct. 2023. doi: 10.1109/TMM.2022.3216951.

[16] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," presented at the IEEE Int. Conf. Comput. Vis., Seoul, Korea (South), Oct. 2019, pp. 6410–6419.

[17] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019. doi: 10.1145/3326362.

[18] Z. H. Lin, S. Y. Huang, and Y. C. F. Wang, "Learning of 3D graph convolution networks for point cloud analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4212–4224, Feb. 2022. doi: 10.1109/TPAMI.2021.3059758.

[19] A. Vaswani *et al.*, “Attention is all you need,” arXiv preprint arXiv:1706.03762, Dec. 2017. doi: 10.48550/arXiv.1706.03762.

[20] H. Zhao, L. Jiang, J. Jia, P. Torr, V. Koltun and I. Labs, “Point transformer,” presented at the IEEE Int. Conf. Comput. Vis., Montreal, QC, Canada, Oct. 2021, pp. 16239–16248.

[21] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou and J. Lu, “Point-BERT: Pre-training 3D point cloud transformers with masked point modeling,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., New Orleans, LA, USA, Jun. 2022, pp. 19291–19300.

[22] X. Lai, L. Jiang, L. Wang, H. Zhao, and X. Qi, “Stratified transformer for 3D point cloud segmentation,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., New Orleans, LA, USA, Jun. 2022, pp. 8490–8499.

[23] L. Wang *et al.*, “SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving,” *Knowledge-Based Syst.*, vol. 259, pp. 110080, Jan. 2023. doi: 10.1016/j.knosys.2022.110080.

[24] M. Xu, R. Ding, H. Zhao, and X. Qi, “PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, Nashville, TN, USA, Dec. 2021, pp. 3172–3181.

[25] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, “Associatively segmenting instances and semantics in point clouds,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 4091–4100.

[26] L. Zhao and W. Tao, “JSNet++: Dynamic filters and pointwise correlation for 3D point cloud instance and semantic segmentation,” *IEEE Trans. Circ. Syst. Video Technol.*, vol. 14, no. 8, pp. 1854–1867, Apr. 2023. doi: 10.1109/TCSVT.2022.3218076.

[27] F. Chen *et al.*, “JSPNet: Learning joint semantic & instance segmentation of point clouds via feature self-similarity and cross-task probability,” *Pattern Recognit.*, vol. 122, no. 1, pp. 108250, Feb. 2022. doi: 10.1016/j.patcog.2021.108250.

[28] Q. Hu *et al.*, “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, Jun. 2020, pp. 11105–11114.

[29] F. Yin, Z. Huang, T. Chen, G. Luo, G. Yu and B. Fu, “DCNet: Large-scale point cloud semantic segmentation with discriminative and efficient feature aggregation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4083–4095, Aug. 2023. doi: 10.1109/TCSVT.2023.3239541.

[30] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, “Graph attention convolution for point cloud semantic segmentation,” presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 10288–10297.

[31] T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv preprint arXiv:1609.02907, Sep. 2017. doi: 10.48550/arXiv.1609.02907.

[32] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, “Linked dynamic graph CNN: Learning through point cloud by linking hierarchical features,” presented at the 27th Int. Conf. Mechatron. Machine Vis. Pra. (M2VIP), Shanghai, China, Nov. 2021, pp. 7–12.

[33] J. Walcza, A. Wojciechowski, P. Najgebauer, and R. Scherer, “Vicinity-based abstraction: VA-DGCNN architecture for noisy 3D indoor object classification,” presented at the Int. Conf. Comput. Sci., Krakow, Poland, 2021, pp. 229–241.

[34] C. Chen, L. Z. Fragonara, and A. Tsourdos, “GAPointNet: Graph attention based point neural network for exploiting local feature of point cloud,” *Neurocomputing*, vol. 438, no. 7553, pp. 122–132, May 2021. doi: 10.1016/j.neucom.2021.01.095.

[35] S. Kim, S. Kim, J. Lee, and H. Yoo, “A low-power graph convolutional network processor with sparse grouping for 3D point cloud semantic segmentation in mobile devices,” *IEEE Trans. Circ. Syst. I: Regular Paper*, vol. 69, no. 4, pp. 1507–1518, Aug. 2022. doi: 10.1109/TCSI.2021.3137259.

[36] Z. Song, L. Zhao, and J. Zhou, “Learning hybrid semantic affinity for point cloud segmentation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4599–4612, Dec. 2022. doi: 10.1109/TCSVT.2021.3132047.

[37] A. Dosovitskiy *et al.*, "An image is worth 16 × 16 words: Transformers for image recognition at scale," presented at the Int. Conf. Learn. Represent., New Orleans, USA, Oct. 2019, pp. 548–558.

[38] H. Vaidwan, N. Seth, A. S. Parihar, and K. Singh, "A study on transformer-based object detection," presented at the IEEE Int. Conf. Intell. Technol., Hubli, India, Jun. 2021, pp. 1–6.

[39] H. Chen *et al.*, "Pre-trained image processing transformer," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Dec. 2021, pp. 12294–12305.

[40] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017. doi: 10.1109/TPAMI.2016.2572683.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," presented at the Int. Conf. Med. Image Comput. Assist. Intervent., Munich, Germany, Oct. 2015, pp. 234–241.

[42] V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Jan. 2017. doi: 10.1109/TPAMI.2016.2644615.

[43] I. Armeni *et al.*, "3D semantic parsing of large-scale indoor spaces," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA, Jun. 2016, pp. 1534–1543.

[44] L. Yi *et al.*, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 6–12, Nov. 2016. doi: 10.1145/2980179.2980238.

[45] S. Qiu, S. Anwar, and N. Barnes, "Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Nashville, TN, USA, Jun. 2021, pp. 1757–1767.

[46] W. Wang and R. Yu, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 2569–2578.

[47] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," presented at the IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Seattle, WA, USA, Jun. 2020, pp. 5588–5597.