Global Context Reasoning for Semantic Segmentation of 3D Point Clouds

Yanni Ma[1], Yulan Guo[1,2]*, Hao Liu[1], Yinjie Lei[3], Gongjian Wen[2]
[1]Sun Yat-sen University, [2]National University of Defense Technology, [3]Sichuan University
mayn3@mail2.sysu.edu.cn, yulan.guo@nudt.edu.cn,
liuh327@mail2.sysu.edu.cn, yinjie@scu.edu.cn, Wengongjian@sina.com

Abstract

Global contextual dependency is important for semantic segmentation of 3D point clouds. However, most existing approaches stack feature extraction layers to enlarge the receptive field to aggregate more contextual information of points along the spatial dimension. In this paper, we propose a Point Global Context Reasoning (PointGCR) module to capture global contextual information along the channel dimension. In PointGCR, an undirected graph representation (namely, ChannelGraph) is used to learn channel independencies. Specifically, channel maps are first represented as graph nodes and the independencies between nodes are then represented as graph edges. PointGCR is a plug-and-play and end-to-end trainable module. It can easily be integrated into an existing segmentation network and achieves a significant performance improvement. We conduct extensive experiments to evaluate the proposed PointGCR module on both indoor and outdoor datasets. Experimental results show that our PointGCR module efficiently captures global contextual dependencies and significantly improve the segmentation performance of several existing networks.

1. Introduction

Semantic segmentation is an important task in computer vision. It has been widely applied in numerous areas such as autonomous driving, virtual reality, mobile robotics, and 3D reconstruction. The task of 2D/3D semantic segmentation is to predict per-pixel/per-point categories of a given scene. Thanks to the advent of depth cameras and laser radars, 3D point clouds have become increasingly accessible. Consequently, the 3D semantic segmentation task has attracted a lot of attention in recent years [23, 16, 38, 30]. However, there is a large performance gap between existing 2D and 3D semantic segmentation algorithms. This is mainly because, it is impractical to directly extend existing 2D segmentation networks to 3D point clouds due to the irregularity and sparsity of point clouds.

Prior works address the unstructured problem of point clouds using different representation methods. Early studies in the area focus on voxel-based and view-based representations [21, 29, 32]. These methods first convert a point cloud into a regular voxel grid or multiple view images, and then apply a Convolutional Neural Network (CNN) on these regular representations. However, due to the information loss of voxel-based and view-based representations, the performance of these methods is limited. An alternative is to directly process unordered point clouds. A pioneering work in this category is PointNet [22], which applies Multi-Layer Perceptrons (MLPs) to obtain order-invariant features of a point cloud. However, PointNet cannot capture local contexts and spatial relationship between features. To address this problem, PointNet++ [23] uses local neighborhoods and hierarchical feature learning to combine multi-scale local structure information. Although PointNet++ can capture local fine-grained and global context information, it still cannot capture spatial distribution and long-term dependency since its features are learned by stacked MLP layers.

It is observed that existing semantic segmentation frameworks usually confuse some particular categories with others. For example, the objects of window and board, the regions of door and wall are usually indistinguishable. To distinguish these confusing categories, it is necessary to learn discriminative feature representations. In recent years, several PointNet based methods have been proposed to address this problem. Two strategies are usually adopted in these methods, including multi-scale context fusion and long-term dependency capturing. For example, Engelmann et al. [8] and [17] aggregated multi-scale spatial contexts in input-level and output-level, while Wu et al. [38] exploited the encoder-decoder structure to fuse high-level and mid-level semantic features. Although the fusion of context features can capture multi-scale information, they still cannot fully explore the global relationship between objects, which is important for semantic segmentation. To capture long-term dependencies, Ye et al. [39] applied a two-direction hierarchical RNN to explore contextual features. However, its performance highly depends on its long-term memorization

capability.

In this paper, we mainly investigate global long-term contextual dependencies to address the aforementioned problems faced by point cloud segmentation. Inspired by the GloRe Unit [6], we propose a module called PointGCR to model global contextual dependencies between channels of 3D point cloud features by learning a graph representation (namely, ChannelGraph). Specifically, it first uses the channel self-attention mechanism to learn point-wise feature similarity between any two channels, and models an initial graph representation of ChannelGraph with its nodes embedded from channels. Then, it learns the dependencies between graph nodes and updates ChannelGraph by passing the relationship information of nodes represented by graph edges. To the best of our knowledge, our method is the first to apply channel-attention with graph convolution on 3D point cloud features. Difference from the GloRe unit [6], our module learns point-wise channel self-attention features on 3D point clouds rather than 2D images. Several improvements are also introduced by our PointGCR module in different aspects such as attention weight calculation and dimension change.

The main contributions of our work are as follows:

- We propose a PointGCR module to model the contextual dependencies between channels using a graph representation and a self-attention mechanism. Our module is lightweight and plug-and-play, it can be conveniently integrated into a point cloud segmentation architecture to significantly improve its performance.
- We introduce a node attention block to embed feature channels as graph nodes, we also present a graph embedding block to learn the dependencies between graph nodes.
- We integrate the proposed PointGCR module into several point cloud segmentation frameworks and perform experiments on three indoor and outdoor scene segmentation datasets. It is demonstrated that our PointGCR module can capture global contextual dependencies and significantly boost the segmentation performance.

2. Related Work

In this section, we will briefly review existing methods in two main areas: point cloud segmentation, and contextual modeling for segmentation.

2.1. Point Cloud Segmentation

Learning discriminative feature representations from point clouds is the foundation for 3D semantic segmentation. A main challenge is how to effectively process irregular and unstructured point clouds using a deep learning approach. According to different representations of point clouds, these methods can be broadly divided into voxel-based, view-based and points-based methods.

Voxel-based Representation. Since point clouds of a 3D scene are unordered and unstructured, we cannot directly extend convolution operations of 2D images to 3D point clouds. Voxelization [18] is proposed to convert irregular point clouds into regular voxel grids, CNN is then applied on this grid data to extract features. Tchapmi et al [29] performed 3D fully convolution on voxels, and then utilized Condition Random Field (CRF) to obtain fine-grained segmentation results. To learn high-resolution representation of 3D data, Riegler et al. [25] utilized unbalanced octrees to divide space hierarchically according to the sparsity of point clouds. Voxelization is an effective way to use CNN, but its performance is limited by the computational complexity and information loss.

View-based Representation. These methods firstly project a 3D point cloud into several a 2D image space, and then use a deep learning model to perform feature learning on images. Su et al. [27] proposed a 3DCNN by projecting a 3D shape onto different viewpoints. VGG-M is then used to learn features of projection views. Kalogerakis et al. [13] obtained a set of shaded images and depth maps of 3D shapes at different viewpoints and scales, and then used a full convolutional network to learn features. The projection used in these methods changes the local and global structure of a 3D shape, which reduces the discriminability of features.

Points-based Representation. PointNet [22] is proposed to directly process irregular point clouds using a deep learning method. It adopts a transformation matrix to keep the point cloud rotation invariant, uses several MLPs to learn point-wise features, and finally employs a symmetric function to obtain global features. PointNet provides an effective way to apply neural networks to point clouds. However, it does not capture the local structure information of a point cloud. To address this issue, PointNet++ [23] is proposed to use local dependencies and hierarchical feature learning to capture multi-scale local structure information. It combines sampling and grouping layers with PointNet to learn local representation. To capture the relationship between individual points, DGCNN [35] is proposed to incorporate edge convolution layers (EdegConv) into the PointNet architecture. It considers the relationship between points as edge features, which are aggregated from points and their k nearest neighbors.

2.2. Contextual Modeling for Segmentation

Image Segmentation. In the area of 2D image segmentation, lots of methods have been proposed based on Fully Convolutional Networks (FCNs). Several methods are also proposed to focus on the capturing and aggregation of con-

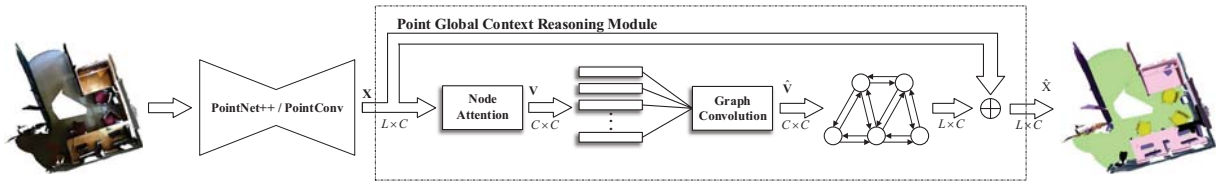

Figure 1. An overview of the proposed PointGCR module. Given a point cloud, we first use an encoder-decoder network (such as PointNet++ or PointConv) to obtain a point-wise feature map. Then, our PointGCR module is appended on top of the feature map to achieve accurate point cloud segmentation.

textual relations for the improvement of feature representations. Deeplabv2 [4] is proposed to explore multi-scale contextual details using Spatial Pyramid Pooling (SPP). The DAG-RNN [26] model is introduced to employ long-range dependencies by incorporating a Recurrent Neural Network (RNN) for relationship learning. A non-local network [34] is also developed to collect long-term relationship for video recognition.

Several works on relationship learning in 3D point clouds are currently available. Engelmann et al. [8] incorporates spatial contexts into segmentation by applying several Multi-Scale (MS) blocks and Consolidation Units (CU). The MS and CU blocks are used to obtain input-level and output-level contexts, respectively. 3DContextNet [40] uses a k-d tree structure to represent point clouds and exploits both local and global contextual cues on this representation. SPG [15] learns the local-to-global contextual information using a Graph Convolution Network (GCN). Here, a GCN uses Gated Recurrent Unit (GRU) and Edge-Conditioned Convolution (ECC) blocks to learn and pass contextual information of global and local features on the whole superpoints graph. DGCNN [35] captures the shape relationship using edge features between individual points. Its edge features are aggregated from points and their corresponding k nearest neighbors. Our method learns the relationship by modeling and reasoning long-term dependencies of high-level feature representations.

Self-attention. Attention mechanism has been used in Natural Language Processing (NLP) to model long-term context dependencies [31]. The self-attention mechanism is an improvement of the attention mechanism, which reduces the dependence on external information and works better in capturing the internal correlation of data or features. It has been currently applied to many visual tasks, including image recognition [34], semantic segmentation [19] and video understanding [37].

Graph-based Representation. Graph representations can be used to model relationships between irregular data. Before the deep learning explosion, long-term dependencies in images or videos have been investigated using graph representations, e.g., through the Conditional Random Field (CRF) method [3]. CRF is usually applied to refine segmentation results. In deep learning architectures, traditional methods such as CRF are gradually replaced by neural networks. In this paper, we introduce the recently proposed GCN [14] to reason global context relationships to improve the performance of point cloud segmentation.

Relational Reasoning. Relational reasoning has demonstrated its strong capacity in many visual tasks such as visual recognition [5], question answering [33] and object detection [36]. It mainly captures the interaction between elements by modeling their dependencies. Several recent works have been proposed to efficiently perform relational reasoning, including the non-local module [34] and the attention module [31][19]. These methods aggregate the information from feature embeddings for all position pairs of the entire input, while the weight for aggregation is constrained by the target task. Our model is associated with a graph convolution, which utilizes the graph structure to represent the context relationship between multiple categories. It automatically generates a neighborhood matrix of the graph convolution operation, and finally passes the information to capture the relationship between these nodes.

3. Method

In this section, we first define the ChannelGraph, and then introduce the nodes and edges of the graph. We also discuss the implementation details of our pointGCR module, and integrate the proposed module into an existing point cloud segmentation network.

3.1. Overview

The goal of this work is to develop a generic module to incorporate relational information with global reasoning. This module is designed to help a network to learn discriminative point cloud features for the improvement of semantic segmentation performance.

High-level features (i.e., the final output) of a feature extraction layer can be considered as category responses. It is believed that interactions exist between these semantic responses, which can be further used to improve segmentation performance. For example, in an indoor scene, tables and chairs, walls and blackboards, windows and doors have particular semantic context dependencies. In an outdoor scene,

cars and roads, roads and plants, people and streets also have particular semantic context dependencies. To capture the semantic context relationship in point clouds, that is, the inter-dependencies between global features, we introduce a point global context reasoning module. We use a graph structure called ChannelGraph to model global feature dependencies and perform global reasoning on the graph representation. Inspired by the channel self-attention mechanism [19], we extract the global feature map as a channel-wise map. Each channel map is used as a graph node, and the dependency between any two channels is used to form a graph edge. Consequently, a graph structure embedding can be constructed.

Using the proposed graph-based representation, all features of the entire feature map can be covered while preserving the spatial relationship. An overview of the proposed method is shown in Fig. 1. More details will be presented in the following sections.

3.2. Point Cloud Segmentation Backbone

Given a large-scale point cloud (e.g., with around 5 billion points), we first randomly down-sample L points from the point cloud. We then use a classical 3D segmentation backbone to extract point-wise features for subsequent label prediction. The backbone model PointConv [38] is a type of encoder-decoder framework for feature extraction and propagation. The backbone model takes a point cloud $\mathbf{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\} \in \mathbb{R}^{L \times C_0}$ as its input, which contains L points with C_0 channels (including position features $\{x, y, z\}$, and color features $\{r, g, b\}$), and produces a $L \times C$ feature map at the end of its last decoder layer.

3.3. ChannelGraph

Given an input feature map $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L\} \in \mathbb{R}^{L \times C}$, where C is the feature dimension (number of channel maps). Our goal is to construct an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the interaction structure of channel maps in a feature map. Note that, each channel map can be consider as a class-specific response. The graph nodes $\mathcal{V}$ represent class-specific responses, while graph edges $\mathcal{E}$ are used to capture long-range contextual dependencies of these semantic responses.

3.4. Graph Node Learning

There are tens of thousands of points in the point cloud of a scene. The large number of points in a point cloud limits the complexity of the feature extraction architecture. Existing methods [22][15] usually use down-sampling or over-segmentation methods to reduce the number of points used for processing. Besides, GCN is a module with a relatively high computational complexity. To enable GCN to learn global context reasoning, it is necessary to embed original features into a relatively small number of graph nodes. Therefore, we reduce the dimension of original feature $\mathbf{X}$ by embedding it into a feature $\mathbf{V}$ with a low computational complexity. These features $\mathbf{V}$ are then used to represent graph nodes.

It is important to design an appropriate approach for the learning of graph node features. Inspired by the self-attention mechanism [31], we use a Node Attention (NA) block to represent the global feature map as a graph structure by embedding these channels into graph nodes. We embed these channels according to their pairwise similarities. The similarity between a pair of channels is formulated as:

$$f(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) \tag{1}$$

where θ and ϕ are two embeddings. We have $\theta(\boldsymbol{x}_i) = W_\theta \boldsymbol{x}_i$ and $\phi(\boldsymbol{x}_j) = W_\phi \boldsymbol{x}_j$, where parameter matrices W_θ and W_ϕ have a size of $C \times C$ and are to be learned during the Back Propagation (BP) process. Using dot multiplication between $W_\phi \boldsymbol{x}_j$ and the transpose of $W_\theta \boldsymbol{x}_i$, a channel attention map $\mathbf{V}$ with a dimension of $C \times C$ is produced. Consequently, the graph nodes of ChannelGraph are obtained and the relationship between responses of different categories is initially modelled. The channel attention map $\mathbf{V}$ is processed with graph convolutions and then has a dot-product with embedding $g(\boldsymbol{x}_j)$ to produce $\hat{\mathbf{V}}$. That is:

$$\hat{\boldsymbol{v}}_i = f(\boldsymbol{x}_i, \boldsymbol{x}_j) g(\boldsymbol{x}_j) \tag{2}$$

Consequently, the channel attention map of size $C \times C$ is transformed to a feature map of size $L \times C$. Here, $g(\boldsymbol{x}_j) = W_g \boldsymbol{x}_j$. Finally, the element-wise sum of the original feature map $\mathbf{X}$ and the new feature map $\hat{\mathbf{V}}$ is obtained through a residual connection [10], resulting in the final feature map:

$$\hat{\boldsymbol{x}}_i = W_v \hat{\boldsymbol{v}}_i + \boldsymbol{x}_i \tag{3}$$

where W_v is a parameter matrix to learn.

Implementation Details. Given the input point features $\mathbf{X} \in \mathbb{R}^{L \times C}$, we feed it into two one-dimensional convolution (Conv1D) layers to obtain two new feature maps $\mathbf{Y} \in \mathbb{R}^{L \times C}$ and $\mathbf{Z} \in \mathbb{R}^{L \times C}$, respectively. Then, we perform dot product between the the transpose of matrix $\mathbf{Y} \in \mathbb{R}^{L \times C}$ and $\mathbf{Z} \in \mathbb{R}^{L \times C}$, resulting in a node attention feature $\mathbf{V} \in \mathbb{R}^{C \times C}$. Through the above transformation operation, the graph nodes of ChannelGraph is embedded. The attended node vector is $\mathbf{V_i} = \{\boldsymbol{v}_{ij} | i, j = 1, 2, \ldots, C\}$, $\mathbf{V_i} \in \mathbb{R}^{C \times C}$, where $\boldsymbol{v}_{ij}$ represents the j-th node feature in $\mathbf{V_i}$. The process of node attention is illustrated in Fig. 2.

3.5. Graph Edge Learning

Once features are embedded into graph nodes, a channel attention map ChannelGraph $\mathbf{V}$ is produced. Each node vector $\boldsymbol{v}_i$ represents a channel map $\boldsymbol{x}_i$. Since we embed these nodes by learning pair-wise similarities, ChannelGraph can be considered as a fully connected graph. The

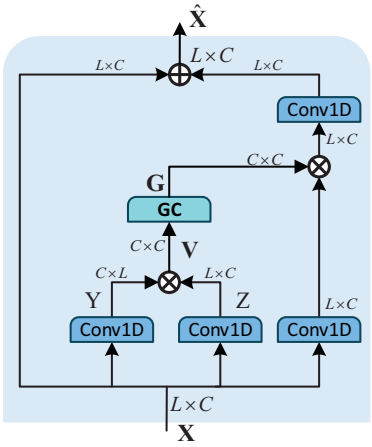

Figure 2. An illustration of PointGCR.

next step is to perform context reasoning to capture the relationship between channels, that is, to learn the graph edges between graph nodes.

In this work, we perform context reasoning on this fully connected graph using GCNs [14]. The graph convolution operation can be represented as:

$$\mathbf{G} = \mathbf{A}\mathbf{V}\mathbf{W}_e, \tag{4}$$

where $\mathbf{V}$ and $\mathbf{G}$ are the input and output graphs with $C \times C$ dimensions, respectively. $\mathbf{A} \in \mathbb{R}^{C\times C}$ represents an adjacency graph, which is used to pass information across graph nodes while reasoning on the ChannelGraph along the node dimension. $\mathbf{W}_e$ represents the edge weight matrix with a dimension of $C \times C$, which is used to update node states while reasoning on the ChannelGraph along the channel dimension.

Implementation Details: The matrices $\mathbf{A}$ and $\mathbf{W_e}$ are initialized randomly and learned during training. We implement the graph convolution by applying two one-dimensional convolution (Conv1D) layers along the node and channel dimensions, respectively. At the first step, we learn the adjacency weight matrix $\mathbf{A}$ by applying a Conv1D along the node dimension to propagate the node information over the ChannelGraph. At the second step, we learn edge weights $\mathbf{W_e}$ by applying a Conv1D along the channel dimension to learn interactive information between the channels within a graph node. Consequently, we can capture both the relationship between channels within each node feature $\boldsymbol{v}_{ij}$ and the inter-dependencies across different nodes. Once the final ChannelGraph representation is obtained, a batch normalization layer and an activate function ReLU are used to improve the training performance.

4. Experiments

In this section, we test our PointGCR module on three 3D point cloud segmentation datasets, including the Stanford Large-Scale 3D Indoor Spaces (S3DIS) [1], ScanNet [7] and Semantic3D [9] datasets. We first briefly describe these datasets, the evaluation metrics and our implementations. Then, we integrate our module into three popular point cloud semantic segmentation frameworks (i.e., PointNet++ [23], PointConv [38]) and PointSIFT [12] to test its performance. In addition, we provide an ablation study of our PointGCR module.

4.1. Experimental Setup

Datasets. We conduct comprehensive evaluation of our PointGCR module on three challenging datasets, i.e., two indoor datasets including S3DIS [1] and ScanNet [7], and one outdoor Semantic3D [9] dataset.

Evaluation Metrics. We quantitatively test the segmentation performance using three metrics: Overall Accuracy (OA), per-class Intersection over Union (IoU), and mean IoU of each class (mIoU). IoU is defined as:

$$IoU = \frac{TP}{TP + FP + FN} \tag{5}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

Implementation Details. In our experiments, we conducted experiments using the tensorflow toolbox and ran programs on an NVIDIA GTX1080 Ti GPU.

Table 1. Semantic segmentation results (%) achieved on the ScanNet dataset. mIoU results of existing methods are from the ScanNet Benchmark Challenge. * denotes the results of existing methods trained by us. The number in each brackets indicates performance improvement (red) or decline (blue).

Method	mIoU (%)	Accuracy (%)
TangentConv [28]	43.8	-
FCPN [24]	44.7	82.6
PointCNN [16]	45.8	85.1
PanopticFusion [20]	52.9	-
PointNet++ [23]	33.9	83.3*
PointNet++-GCR (Ours)	37.8 (3.9)	84.7 (1.4)
PointConv [38]	58.2*	-
PointConv-GCR (Ours)	60.8 (2.6)	-
PointSIFT [12]	41.0*	84.2*
PointSIFT-GCR (Ours)	42.7 (1.7)	85.3 (1.1)

4.2. Results on the ScanNet Dataset

The ScanNet [7] dataset is a large-scale 3D indoor dataset, which contains 1513 scanned and reconstructed indoor scenes. For the semantic segmentation task, 20 categories are provided for evaluation. In our experiments, 8192 points were randomly sampled from each point cloud for training.

Performance. We integrated our PointGCR module into PointNet++ [23], PointConv [38] and PointSIFT

Table 2. Semantic Segmentation results (%) achieved on the S3DIS dataset (*Area-5*). IoU results of existing methods are from [29], [11] and [39]. The number in each brackets indicates performance improvement (red) or decline (blue).

Method	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter	**mIoU**
PointNet [22]	88.80	97.33	69.80	0.05	3.92	46.26	10.76	52.61	58.93	40.28	5.85	26.38	33.22	41.09
SEGCloud [29]	90.06	96.05	69.86	0.00	18.37	38.35	23.12	75.89	70.40	58.42	40.88	12.96	41.60	48.92
RSNet [11]	93.34	98.36	79.18	0.00	15.75	45.37	50.10	65.52	67.87	22.45	52.45	41.02	43.64	51.93
SPGraph [15]	91.50	97.91	75.95	0.00	14.23	51.29	52.26	86.38	77.43	65.51	40.44	7.23	50.77	54.68
PointNet++ [23]	90.79	96.45	74.12	0.02	5.77	43.59	25.39	69.22	76.94	21.45	55.61	49.34	41.88	50.04
PointNet++-GCR (Ours)	90.71	96.13	74.85	0.10	16.09	50.17	32.29	68.95	78.09	41.26	60.69	53.82	43.76	54.38
	(-0.08)	(-0.32)	(0.73)	(0.08)	(10.32)	(6.58)	(6.90)	(-0.27)	(1.15)	(19.81)	(5.08)	(4.48)	(1.88)	(4.34)
PointConv [38]	92.03	97.79	73.79	0.00	3.31	43.67	23.02	69.90	76.80	32.43	54.11	44.50	43.10	50.34
PointConv-GCR (Ours)	93.12	98.02	76.44	0.00	17.24	42.46	27.10	74.32	83.14	18.95	62.18	39.28	49.22	52.42
	(1.09)	(0.23)	(2.65)	(0.00)	(13.93)	(-1.21)	(4.08)	(4.42)	(6.34)	(-13.48)	(8.07)	(-5.22)	(6.12)	(2.08)
DGCNN [35]	92.42	97.46	76.03	0.37	12.00	51.59	27.01	64.85	68.58	7.67	43.76	29.44	40.83	47.08
DGCNN-GCR (Ours)	91.73	97.66	75.98	0.00	15.09	53.43	21.19	67.42	69.08	19.44	51.92	39.06	44.39	49.72
	(-0.69)	(0.20)	(-0.05)	(-0.37)	(3.09)	(1.84)	(-5.82)	(2.57)	(0.50)	(11.77)	(8.16)	(9.62)	(3.56)	(2.64)

[12] networks, resulting in three new networks, namely, PointNet++-GCR, PointConv-GCR and PointSIFT-GCR. We further compare them to existing methods including PointNet++ [23], PointConv [38], PointSIFT, PointCNN [16], and TangentConv [28]. The mIoU results achieved on the ScanNet dataset are shown in Table 1. It can be seen that, the networks integrated with our PointGCR module obtain a significant improvement as compared to their baseline networks (i.e., 3.9 % for PointNet++, 2.6 % for PointConv blue and 1.7% for PointSIFT in terms of mIoU). Compared to FCPN [24], PointCNN [16] and PanopticFusion [20], the network PointConv-GCR with our PointGCR module achieves the best segmentation performance.

4.3. Results on the S3DIS Dataset

The S3DIS [1] dataset contains 3D points of 6 areas with 271 different rooms, including hallways, conference rooms, and lounges. This dataset is acquired with Matterport scanners. Each point is annotated with one of 13 semantic class labels, including chair, table, wall, ceiling, and clutter. In our experiments, we performed 6-fold cross validation over 6 areas. For fair comparison, we choose Area 5 as the test set and train our module on the remaining 5 areas.

Performance. We integrated our PointGCR module into DGCNN [35], PointNet++ [23] and PointConv [38] networks, and also compare their results to PointNet [22], SEGCloud [29], 3DContextNet [40], DGCNN [35], and SPGraph [15]. Comparative results are listed in Table 2. It can be seen that, the best mIoU performance achieved with our PoinGCR module is 54.38%, which is increased by 4.34% as compared to the baseline network PointNet++. Specifically, PointNet++-GCR achieves performance improvement on 10 categories out of all 13 categories as compared to its baseline PointNet++. Among these categories, bookcase and column are very hard to identify by existing methods. However, with our PoinGCR module, the IOU performance of PointNet++ is improved by 19.81% and 10.32% on these two categories, respectively. Besides, the IOU performance of PointNet++ on door and window has also been improved by over 6.5%. Further, our module also brings a significant performance improvement to the baseline methods DGCNN and PointConv. These results clearly demonstrate that our module can implicitly learn the relationship between different categories.

4.4. Results on the Semantic3D Dataset

The Semantic3D [9] dataset is the largest publicly available 3D outdoor dataset, which contains more than 40 million points acquired from urban and rural scenes. Each point has RGB and intensity values and is labeled with one of the 8 semantic categories: man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artifacts, and cars.

Performance. We conducted experiments on the reduced-8 subset, which is a reduced version of the Semantic3D dataset. We also compare our PointNet++-GCR and PointConv-GCR to several existing methods including PointNet [22], SEGCloud [29], 3DContext [40], DGCNN [35] and SPGraph [15], as shown in Table 3. It can be seen that the PointNet++ framework integrated with our PointGCR module outperforms its baseline by 4.7% in terms of mIoU and 5.8% in terms of OA. PointGCR also improves the performance of the baseline PointConv framework. Specifically, PointConv with our PointGCR module obtains performance improvements in 6 out of 8 categories. The hard scape and low vegetation categories are improved by about 10.5% and 3.7% in terms of IoU. From Sections 4.2-4.4, it can be concluded that our relationship learning module has a good generalization capability and works well on different indoor and outdoor datasets.

Table 3. Semantic segmentation results (%) achieved on the Semantic3D dataset (*semantic-8 challenge*). IoU data of existing methods are from the Semantic3D online evaluation website. The number in each brackets indicates performance improvement (red) or decline (blue).

Method	man-made terrain	natural terrain	high vegetation	low vegetation	buildings	hard scape	scanning artefacts	cars	**mIoU**	**OA**
FCNVoxNet	6.6	27.2	58.0	36.4	80.9	28.3	9.5	50.9	37.2	52.3
DeepSegNet [9]	89.4	81.1	59.0	44.1	85.3	30.3	19.0	5.0	51.6	88.4
SnapNet [2]	89.6	79.5	74.8	56.1	90.9	36.5	34.3	77.2	67.4	91.0
SPGraph [15]	91.5	75.6	78.3	71.7	94.4	56.8	52.9	88.4	76.2	92.9
PointNet++ [23]	81.9	78.1	64.3	51.7	75.9	36.4	43.7	72.6	63.1	85.7
PointNet++-GCR (Ours)	92.3	79.6	60.3	59.2	92.2	34.3	42.4	82.3	67.8	91.5
	(10.4)	(1.5)	(-4.0)	(7.5)	(16.3)	(-2.1)	(-1.3)	(9.7)	(4.7)	(5.8)
PointConv [38]	92.2	79.2	73.1	62.7	92.0	28.7	43.1	82.3	69.2	91.8
PointConv-GCR (Ours)	93.8	80.0	64.4	66.4	93.2	39.2	34.3	85.3	69.5	92.1
	(1.6)	(0.8)	(-8.7)	(3.7)	(1.2)	(10.5)	(-8.8)	(3.0)	(0.3)	(0.3)

4.5. Visualization

A visualization of segmentation results achieved by PointConv and PointConv-GCR is shown in Fig. 4. Note that, red boxes are used to highlight the major differences between these segmentation results. It can be seen that our PointConv-GCR can accurately segment detailed structures and contours. That is mainly because our module can learn discriminative features by utilizing the attention mechanism and GCN. For example, PointConv incorrectly classifies some points on table as desk and classifies several points on cabinet as counter on Scene0095_01. On Scene0100_00, PointConv predicts curtain as door and classifies the majority of the cabinet as counter. This is because, PointConv is unable to effectively capture long-term context dependencies and to classify isolated points in the scene from a global perspective. As shown in the last two rows in Fig. 4., we can find that PointConv always confuses the points on board and windows (which are attached to the wall) on the scene conference_3. On office_14, PoinConv produces a large number of mis-classified points on the board. However, with our PointGCR module, a more complete and smoother results can be obtained, especially for these objects attached to the wall.

Table 4. Ablation study on two major components of the PointGCR module on S3DIS.

Baseline	NA	GCN	2GCN	SE-net	S3DIS [1]
PointNet++ [23]					50.04
	✓				51.69
	✓	✓			54.38
	✓		✓		51.79
				✓	51.88
PointConv [38]					50.34
	✓				50.54
	✓	✓			52.42
	✓		✓		52.46
				✓	51.30

Table 5. Ablation study on the depth of baseline networks.

Baseline	two layers	four layers	PointGCR	S3DIS [1]
PointNet++ [23]	✓			42.16
	✓		✓	46.70
		✓		47.59
		✓	✓	49.16
PointConv [38]	✓			41.74
	✓		✓	47.66
		✓		47.31
		✓	✓	52.41

4.6. Ablation Study

Ablation study on components of PointGCR . We apply our module on top of PointNet++ and PointConv networks to capture long-term dependencies. To verify the effectiveness of major components in our module, we perform experiments on their combinations. The results on the S3DIS dataset are summarized in Table 4.

It is clear that, the graph reasoning block improves the performance significantly. Compared to the baseline PointNet++, using the GCN reasoning block obtains an mIoU of 54.38%, which brings an increase of 4.34%. However, using two GCN reasoning blocks only produces about 1.75% increase in mIoU. That means, multiple stacking of GCN blocks cannot improve the performance. In addition, if only the similarity relationship modeling module is used without GCN reasoning, mean IoU can also be improved by about 1.65%. Similar conclusions can also be observed for PointConv. These results show that relationship reasoning makes a major contribution to the improvement of segmentation performance.

In addition, we insert SE-net module into PointNet++ and PointConv to test their performance on the S3DIS dataset. The SE-net module brings improvements of 1.84% on PointNet++ and 0.96% on PointConv. In contrast, the improvement brought by our module on PointNet++ and PointConv are 4.34% and 2.08%, respectively.

Ablation study on the depth of baseline networks. To

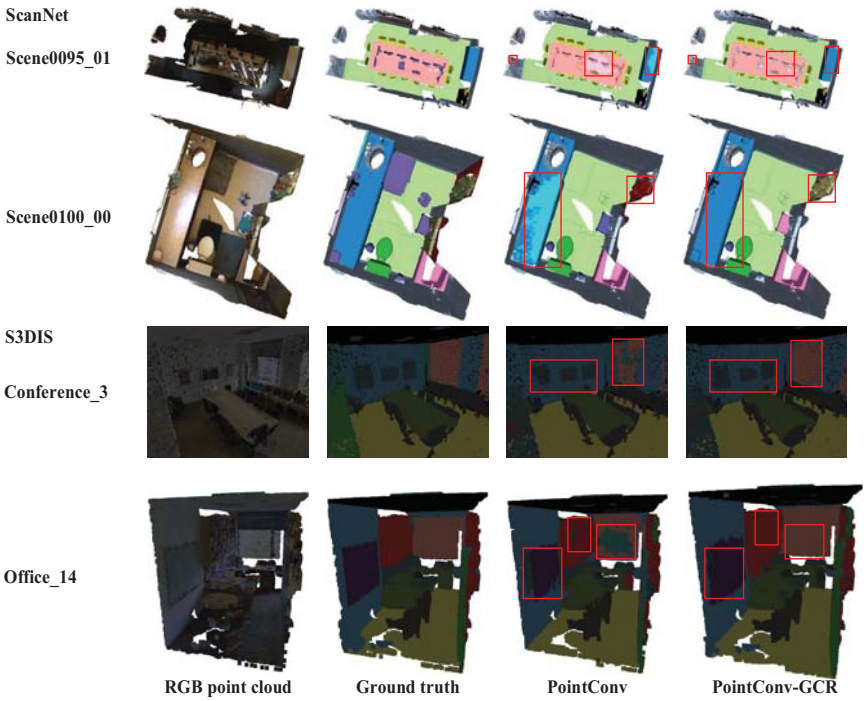

Figure 3. Semantic segmentation results obtained by PointConv [38] and its PointGCR variant on the ScanNet [7] dataset. From left to right: RGB point clouds, groundtruth, PointConv, PointConv-GCR. Comparative results between PointConv and PointConv-GCR are highlighted in red boxes.

Table 6. Parameters and model sizes of different networks.

Methods	PointNet++	PointNet++-GCR	PointConv	PointConv-GCR
Parameters	1.41M	1.55M	3.93M	4.07M
Model Size (MB)	11.4	13.3	259.9	261.9

further demonstrate the effectiveness of PointGCR, we conduct ablation experiments using PointNet++ and PointConv frameworks. Note that, PointNet++ consists of 4 feature extraction layers followed by 4 propagation layers. We removed the last feature extraction layer and the first propagation layer, the results are summarized in Table 5. The performance of PointNet++ is reduced by removing these two layers. However, the mIoU performance is improved (i.e., 1.57% for PointNet++ and 5.10% for PointConv) after appending our PointGCR module. Further, when the last 2 feature extraction layers and the first 2 propagation layers are removed, a significant gain in mIoU (i.e., 4.54% for PointNet++ and 5.92% for PointConv) is also achieved. These results demonstrate that our PointGCR module significantly improves point cloud segmentation performance by reasoning long-term relationship.

Parameters and model size. We summarize the parameters numbers and model sizes of PointNet++, PointConv and their PointGCR variants in Table 6. This further demonstrates that our PointGCR module can improve the segmentation performance without significant increase in computational complexity.

5. Conclusion

In this paper, we have proposed a PointGCR module for semantic segmentation of point clouds. We define a graph representation ChannelGraph to model the global long-term contexts for relational reasoning. We embed the nodes of ChannelGraph using channel attention, and learn the edge weights by performing graph convolutions along the channel dimension. Our PointGCR module is plug-and-play and end-to-end trainable. Extensive experiments have been conducted on three different datasets. Experimental results show that, our PointGCR module can introduce significant and consistent improvements to existing point cloud segmentation networks.

6. Acknowledge

This work was partially supported by the National Natural Science Foundation of China (No. 61972435, 61602499), Natural Science Foundation of Guangdong Province (2019A1515011271), Fundamental Research Funds for the Central Universities (No. 18lgzd06), Shenzhen Technology and Innovation Committee, and the Key Research and Development Program of Sichuan Province (2019YFG0409).

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016.

[2] A. Boulch, B. Le Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2017.

[3] S. Chandra, N. Usunier, and I. Kokkinos. Dense and low-rank gaussian CRFs using deep embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *Computer Research Repository*, abs/1606.00915, 2016.

[5] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta. Iterative visual reasoning beyond convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Y. Chen, M. Rohrbach, Z. Yan, S. Yan, J. Feng, and Y. Kalantidis. Graph-based global reasoning networks. *CoRR*, abs/1811.12814, 2018.

[7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017.

[8] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[9] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3D.net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Computer Research Repository*, abs/1512.03385, 2015.

[11] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2626–2635, 2018.

[12] M. Jiang, Y. Wu, and C. Lu. Pointsift: A sift-like network module for 3D point cloud semantic segmentation. *CoRR*, abs/1807.00652, 2018.

[13] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3D shape segmentation with projective convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] T. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. 09 2016.

[15] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.

[16] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 820–830, 2018.

[17] Y. Ma, Y. Guo, Y. Lei, M. Lu, and J. Zhang. 3DMAX-Net: A multi-scale spatial contextual network for 3d point cloud semantic segmentation. pages 1560–1566, 08 2018.

[18] D. Maturana and S. Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[19] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. *arXiv preprint arXiv:1903.01177*, 2019.

[21] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017.

[24] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018.

[25] G. Riegler, A. Osman Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3577–3586, 2017.

[26] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Scene segmentation with dag-recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1480–1493, 2018.

[27] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[28] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2018.

[29] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In *Proceeding of International Conference on 3D Vision (3DV)*, pages 537–547. IEEE, 2017.

[30] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. KPConv: Flexible and deformable convolution for point clouds. *arXiv preprint arXiv:1904.08889*, 2019.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.

[32] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, pages 88 – 100, 2015.

[33] S. Wang, W. Zhao, Z. Kou, and C. Xu. How to make a BLT sandwich? learning to reason towards understanding web instructional videos. *Computer Research Repository*, abs/1812.00344, 2018.

[34] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[36] S. Woo, D. Kim, D. Cho, and I. S. Kweon. Linknet: Relational embedding for scene graph. In *Advances in Neural Information Processing Systems 31*, pages 560–570. 2018.

[37] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[38] W. Wu, Z. Qi, and L. Fuxin. PointConv: Deep convolutional networks on 3D point clouds. *arXiv preprint arXiv:1811.07246*, 2018.

[39] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[40] W. Zeng and T. Gevers. 3DContextNet: K-d tree guided hierarchical learning of point clouds using local contextual cues. *Computer Research Repository*, 2017.