3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation

Xiaoqing Ye[1,2][0000−0003−3268−880X], Jiamao Li[1][0000−0002−7478−4544], Hexiao Huang[3], Liang Du[1,2], and Xiaolin Zhang[1]

[1] Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China
[2] University of Chinese Academy of Sciences, Beijing, China
{qingye,jmli}@mail.sim.ac.cn
[3] School of Science and Technology, Shanghai Open University, Shanghai, China

Abstract. Semantic segmentation of 3D unstructured point clouds remains an open research problem. Recent works predict semantic labels of 3D points by virtue of neural networks but take limited context knowledge into consideration. In this paper, a novel end-to-end approach for unstructured point cloud semantic segmentation, named 3P-RNN, is proposed to exploit the inherent contextual features. First the efficient pointwise pyramid pooling module is investigated to capture local structures at various densities by taking multi-scale neighborhood into account. Then the two-direction hierarchical recurrent neural networks (RNNs) are utilized to explore long-range spatial dependencies. Each recurrent layer takes as input the local features derived from unrolled cells and sweeps the 3D space along two directions successively to integrate structure knowledge. On challenging indoor and outdoor 3D datasets, the proposed framework demonstrates robust performance superior to state-of-the-arts.

Keywords: 3D semantic segmentation · Unstructured point cloud · Recurrent neural networks · Pointwise pyramid pooling

1 Introduction

Scene understanding has been extensively studied due to its critical role in autonomous driving, robot navigation, augmented reality and 3D reconstruction. Despite of the tremendous progress made in the field of semantic segmentation with the help of deep learning strategies, most approaches cope with 2D images [1–3], whereas 3D semantic segmentation of unstructured point clouds remains a challenging problem due to its large-scale point data, irregular shape and non-uniform densities.

Previous learning-based attempts mainly focus on regularizing input point cloud shapes, so as to draw on the experience of 2D semantic segmentation networks. For example, points are first voxelized by volumetric occupancy grid representation and 3D Convolutional Neural Networks (CNN) are employed to

learn voxel-level semantics. Due to the sparsity of point clouds, the voxelization is inefficient and fine-details are missed to avoid high computation cost. Besides, the accuracy is limited because all points within the same voxel are assigned with the same semantic label. To make use of 2D frameworks, snapshots of 2D images taken at multi-views of 3D space are also learned, however, the reprojection back to 3D space is also a nontrivial problem.

The first pioneer work PointNet that directly operates on 3D point cloud is recently proposed [4]. Without the transformation to voxels, the architecture preserves inherent information within the raw points to predict point-level semantics. PointNet takes advantage of Multilayer Perceptron (MLP) to learn high-dimensional local features for each point individually. Then the local features are aggregated by symmetric max pooling to yield global feature, which is invariant against the permutations of points. However, the architecture has two limitations that restrict its performance to larger and more complicated point clouds. For one thing, only the pointwise features along with the pooled global features are integrated, failing to capture local structures represented by neighboring points. For another, a point cloud is first subdivided into small volumetric blocks and each block is predicted independently without any connection. In consequence, the overall accuracy of PointNet is limited in complicated scenes.

To tackle the first problem, we adopt the one-stride pyramid pooling to aggregate multi-scale neighboring knowledge due to its nonparametric feature and efficiency in enlarging receptive field. Instead of replicating the global pooled features for all points as PointNet does, we perform pointwise pooling and each point is represented by particular pyramid local features. Note that we employ one-stride multi-window pooling rather than multi-stride fixed-window pooling units, in view of preserving fine-grained details. With regard to the second problem, we further integrate long-distance context by means of a two-step hierarchical RNN model. Specifically, the point cloud is first subdivided into partially overlapped blocks along the two horizontal directions, namely, x and y, respectively. The first set of RNNs are applied to the blocks along the x-direction, which updates the state and output according to the long-dependency neighboring blocks. Next, the features derived from the first RNN set are further fed into another set of RNNs along y-direction to integrate relevant context across the horizonal dimensions. This is because adjacent objects or large objects indicate some inherent contextual connection, which helps to solve the ambiguity. For example, chairs are often near the table, and windows are generally inside the wall. Experimental results on the challenging point cloud datasets reveal that our strategy largely improves the accuracy for 3D semantic segmentation.

To sum up, the main contributions of our work are as follows:

- We propose a novel end-to-end framework for unstructured point cloud semantic segmentation, incorporating local spatial structures as well as long-dependency context. The pointwise pyramid pooling (3P) module increases the overall accuracy at negligible extra overhead.
- We introduce a two-direction hierarchical RNN model to learn long-range spatial context and inherent connection for pointy cloud semantic segmenta-

tion. To the best of out knowledge, this is the first time that a two-direction tactic RNN model is investigated to perform 3D semantic segmentation task.
- Our framework presents new state-of-the-art performance on indoor and outdoor 3D semantic datasets.

2 Related Work

Traditional point cloud semantic segmentation algorithms largely rely on hand-crafted features and well-designed optimization approaches, sometimes preprocessing and post-processing strategies are required to achieve better performance. In this work we mainly focus on the review of deep learning strategies that are more related to our work.

Motivated by the large collections of 3D scene datasets such as indoor datasets NYU V2 [5], S3DIS [6], ScanNet [7] and outdoor datasets Semantic.3D [8], KITTI [9], vKITTI [10], great progress of point cloud processing has been made in recent years. However, due to the irregularity and inconsistent point densities of 3D geometric data, classic CNNs are unable to directly deal with point cloud data inputs. As a result, some alternatives have been tailored to the problem.

Voxel-based 3D CNNs: In order to represent 3D geometric shape, a point cloud is first converted into regular volumetric occupancy grids and then trained by 3D CNNs to yield voxel-level predictions [7, 11, 12]. However, uniform 3D arrangement is inefficient because of the sparsity of 3D data. Besides, due to the expensive computation of 3D convolutions than 2d ones, the voxel size is constrained to a relatively small size and accordingly, it is challenging for such architectures to be extended to large-scale point clouds. Unbalanced octrees were exploited to tackle the sparsity problem, which allowed to train 3D CNNs at higher resolutions [13]. SEGCloud subsampled large cloud into voxels and post-processed by trilinear interpolation and conditional random field [14]. However, all the voxel-based methods fail to achieve point-level accuracy.

Multi-view CNNs: An alternative is to project the 3D point cloud into 2D image rendering of multiple views and apply well-engineered 2D CNNs [2, 15–17] to jointly classify them. Multi-view CNN (MVCNN) [18] was designed on top of image-based classification networks, which integrated the views taken around a 3D meshed object through view pooling. Multi-resolution filtering in 3D space was introduced to further improve the performance of multi-view CNNs [19]. SnapNet [20] generated snapshots around the 3D scene in the form of RGB and depth image pairs and applied 2D neural networks to process them separately. SnapNet-R [21] improved the baseline work SnapNet by directly processing RGB-D snapshots in numerous views for dense 3D point labeling. However, 2D snapshots break the inherent relationship within 3D data and thus fails to exploit the full power of 3D spatial context. Besides, it is not direct enough and requires extra 2D to 3D re-mapping.

Deep learning on unordered point sets: PointNet [4] was the first architecture that directly worked on raw point sets to produce per-point classes. Pointwise features and the aggregated global feature were concatenated to make

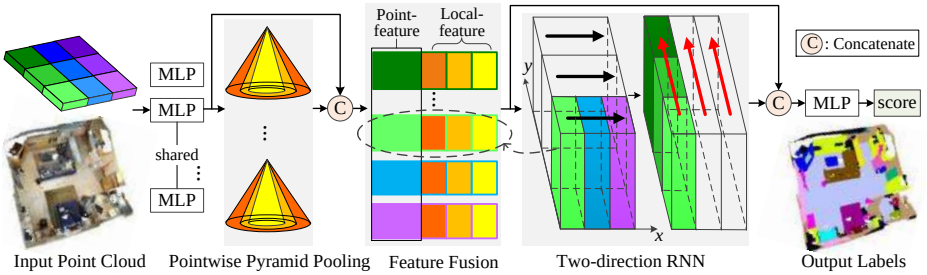

Fig. 1. Overview of the proposed approach. The architecture takes as input the unstructured point cloud and outputs pointwise semantic labels. Point-features and local cell features are concatenated and passed through the two-direction RNN module along x and y. The output of the first RNNs (black arrowed) are reorganized and fed to the next RNNs (red). For details of pointwise pyramid pooling, see Fig. 2.

pointwise predictions. However, the absence of neighboring context structure limited the segmentation performance on complicated scenes. To overcome this drawback, the hierarchical approach PointNet++ [22] was designed to better capture local structures as well as generalize to variable densities. Also inspired by the recent PointNet work, multi-scale windows or neighboring cell positions were exploited to incorporate neighborhood knowledge [23]. Due to the efficiency in extracting point-features, the PointNet framework was further extended to learn local shape properties [24] as well as predict 3D Object Detection from RGB-D Data [25]. Graph neural networks (GNNs) were undertaken to spread contextual information on top of 3D points [26]. Nevertheless, each node in the k-nearest neighbor graph requires additional 2D appearance features for initialization. A very recent GNN work captured the structure of point clouds by superpoint graph, which partitioned various objects into simple shapes and assigned segmentation labels to each part as a whole [27]. The GNNs iteratively updated the node state by message propagation over neighboring nodes. In terms of realization, the dynamic models can be implemented as Recurrent Neural Network.

3 3D Recurrent Neural Networks with Context Fusion

The proposed framework takes inspiration from PointNet [4], which is briefly reviewed in the following part. The baseline is extended with two distinctive improvements to learn local and long-dependency spatial context for better performance. For one thing, a pointwise pyramid pooling module is proposed to learn multi-scale neighboring context. Though simple, it is more efficient than multi-scale input context aggregation in [22, 23] because of the non-parametric pooling units. For another, long-range context is exploited by the two-direction RNN model, which enables the network to learn spatial context in large-scale point clouds. An overview of our approach is presented in Fig. 1.

3.1 Review of PointNet

In the vanilla PointNet work, given a set of unstructured 3D points $\{p_1, p_2, ..., p_N\}$ with $p_i \in R^d$, they are first divided into small overlapped 3D blocks and each block is handled independently. Multi-layer perceptron (MLP) is exploited to learn high-dimensional spatial encoding of each point, i.e., the pointwise features. The block feature is obtained by aggregating pointwise features by a single max pooling within the same block, i.e., the global feature. The global feature is then duplicated for N tiles and concatenated with the corresponding pointwise features to yield the final prediction score. The operation of PointNet can be represented by

$$F\left(p_1, p_2, ..., p_N\right) = \underset{i=1,...,N}{MLP}\left(C\{maxpool\left\{\underset{i=1,...,N}{MLP}\left(p_i\right)\right\}, \underset{i=1,...,N}{MLP}\left(p_i\right)\}\right) \quad (1)$$

where C denotes the concatenation operation, indices of points' index from 1 to N below MLP denotes the MLP operation is pointwise. Nevertheless, the semantic segmentation performance is limited by a lack of neighboring structure and long-range context.

3.2 Pointwise Pyramid Pooling

In this work, we propose a simple but efficient strategy to capture local neighboring context robust to densities. Owing to the brutal global max pooling in PointNet, it is prone to missing fine details and causing ambiguity. Multi-scale grouping and multi-resolution grouping of points within a radius to the certain point are leveraged to learn the structure information in [22]. Alternatively, multi-scale and multi-grid input context are employed to compute block-features in [23]. Both of these strategies capture multi-scale local structures at the expense of indirected and complex fusion strategy, as well as extra computation.

Different from classic 2D pooling unit that employs various strides, we adopt a pointwise pyramid pooling (3P) module with multi-size pooling windows inspired by [28]. This is because pooling module with stride larger than one could bring about a loss of resolution and hampers the accuracy of dense prediction. In specific, given a set of unordered points, we first divide the whole 3D space into blocks of 1.5m $\times$ 1.5m along the ground plane. Each block is extended to cover the whole room height. Pyramid pooling is done at neighborhoods with different number of points. Rather than searching k-nearest neighbors for each point, instead we adopt an approximate but more efficient way leveraging multiple scales of cuboids. In other words, we further subdivide each block into smaller cuboids at different scales. At each scale, one-stride max pooling module with corresponding pooling window size is employed. For example, if window size is N, we randomly select N points within the corresponding cuboid for max pooling. The 3P pooling can be denoted as

$$P\left(p_1, p_2, ...p_N\right) = \left[\underset{p=p_1,...,p_N}{maxpool}\left(f, k_1\right), ..., \underset{p=p_1,...,p_N}{maxpool}\left(f, k_m\right)\right] \quad (2)$$

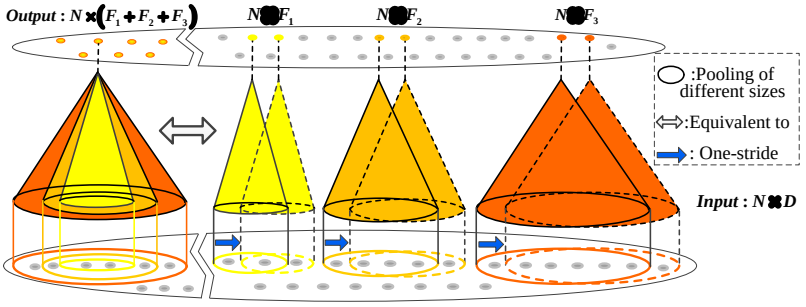

Fig. 2. Pointwise pyramid pooling. Given $N \times D$ input features, each pooling outputs features with the same number of input points (one-stride) and is then concatenated.

where k_i denotes one-stride pooling window size. f represents high-level features learned by MLP. Notably compared to Eq. 1, we move the representation of point set range $\{p = p_1, p_2, ..., p_N\}$ from inside the max-pooling operation to outside, since Eq. 1 attains a single output for all points within the block, whereas ours in Eq. 2 yields the same size of output features as input vectors, namely, pointwise. In our architecture, the adopted window size are N, $N/8$ and $N/64$, respectively.

The attained coarse-to-fine pooled features are then integrated by a single convolution layer for the subsequent RNN stage. The sketch map of our one-stride 3P module is depicted in Fig. 2. Thanks to its nonparametric feature and efficiency in enlarging the receptive field, it is able to achieve an optimized trade-off between accuracy and cost.

3.3 Recurrent Neural Networks for Context Ensemble

Impelled by the successful application of RNN models in 2D semantic segmentation [29–32], we introduce our two-direction hierarchical RNN model for 3D point cloud labeling to make use of the long-range spatial dependency. The whole 3D space is split into uniformly-spaced blocks along x and y direction on the ground plane, i.e., $L_x \times L_y$, respectively. The space along up-right axis is kept undivided due to its high sparsity and coherency in vertical direction. The detailed pipeline is depicted in Fig. 3.

The original input point-features and the output of the pyramid pooling module are first concatenated to be taken as input of the RNN model. In essence, the extra concatenation layer can be omitted if we add a pooling unit with window size equaling to one in the previous pointwise pyramid pooling module. In Fig. 1, we still present the concatenation operation for clear depiction.

The pipeline involves two stages. In the first stage (arrowed in black in Fig. 3), we only consider spatial connection along x direction by coupling L_x small blocks within the same y index as a whole. Note that the operation of each recurrent group is independent and can be realized in parallel. The features derived from point features concatenated with pooled features within each block are unrolled

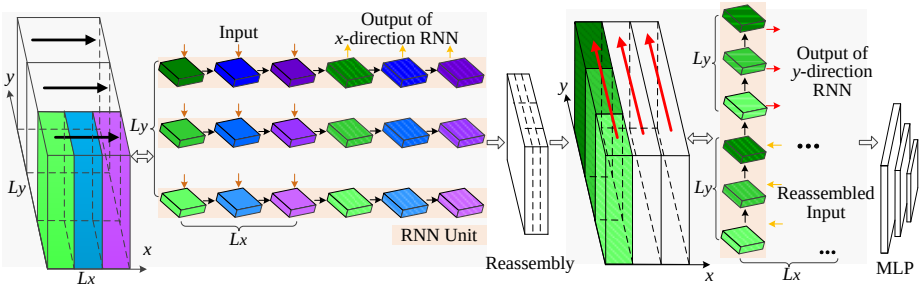

Fig. 3. The pipeline of our two-direction hierarchical RNN module. Output of the first x-direction RNN is reassembled for the input of the second y-direction RNN.

to form a sequence of the corresponding RNN cells. At each time step (corresponding to each small block along the same y index), every RNN takes the concatenated block features as input and based on the previous state from its preceding neighboring blocks to update the new state as follows:

$$O_{i,j}, S_{i,j} = f\left(x_{i,j}, S_{i-1,j}\right) \quad for\ i = 1, ..., L_x \tag{3}$$

where $x_{i,j}, S_{i-1,j}$ denote the current input and the previous state of a certain block, and $O_{i,j}, S_{i,j}$ represent the output and the updated state, respectively. Since the recurrent layers are operated in a many-to-many mode, it is indicated that the complete output will be returned only after the entire input sequence has been passed through the recurrent layer, which is capable of learning long-range dependencies along the x direction. In specific, each one-dimensional recurrent layer can be simply implemented as multi-LSTMs [33] or GRUs [34], in our work multi-LSTM is adopted.

After all the points are swept over the x dimension, the attained brand-new features for each small block are served as input of the following stage. In the second stage, we recompose the features to consider spatial relevance along y direction. Specifically, for each recurrent layer along y-dimension, block features of the same x index are unrolled and composed to form a new sequence. In other words, there are L_x RNN layers with each consisting of L_y timestamps in the second stage. Analogously, at each time step, we proceed reading one element and update the state asynchronously as follows:

$$O_{i,j}, S_{i,j} = f\left(\tilde{x}_{i,j}, S_{i,j-1}\right) \quad for\ j = 1, ..., L_y \tag{4}$$

where $\tilde{x}_{i,j}$ is the updated features derived from the first stage. In other words, the resulted features from x-direction RNNs are taken as input for the y-direction operation.

After both directions along the ground plane have been processed, we obtain the updated features originated from integrating local and long-range spatial context knowledge. Especially, we do not break the inherent connection within each block. Instead, our model learns to share long-range knowledge by propagating neighboring features along the two directions hierarchically. Note that one

can also stack more recurrent layers for processing additional directions, we only choose the mentioned two directions in view of memory and speed. The output features of the RNN-based model are then concatenated with the original input features, including the pointwise features and local pooled features, to predict the final label for each point.

4 Experimental Results

4.1 Dataset and Evaluation Criteria

In this section, evaluations are mainly carried out on the following challenging datasets: Stanford Large-Scale 3D Indoor Spaces (S3DIS) [6], ScanNet [7], as well as outdoor vKITTI [10], KITTI Raw [9] and 3DRMS Challenge [35].

S3DIS dataset is an indoor 3D point cloud dataset that includes six large-scale indoor areas that originate from three different buildings, totally covering over 6,000 m^2 and involving thirteen semantic classes.

ScanNet dataset contains over 1500 scanned 3D indoor scenes and 21 semantic classes. Experiment settings are borrowed from [7] to split the dataset into $4:1$ for training and testing, respectively.

vKITTI dataset is a synthetic large-scale outdoor dataset imitating the real-world KITTI dataset, with 13 semantic classes in urban scenes. The annotated point clouds are obtained by projecting 2D semantic labels into 3D space.

KITTI Raw dataset contains sparse Velodyne LiDAR point clouds without color information. Due to a lack of semantic ground truth labels, it can not be employed for supervised training. However, the density is comparable to vKITTI and we leverage it for generalization validation.

For evaluation, we report quantitative and qualitative results on indoor and outdoor datasets. The evaluation metric used in our experiments are: mean intersection over union over all classes (mIoU), per-class IoU, mean per-class accuracy (mAcc) and overall accuracy (OA). In specific, IoU can be computed as

$$IoU = \frac{TP}{(T + P - TP)} \tag{5}$$

where TP is the number of true positives, T is the number of ground truth positive samples, P is the number of predicted positives belonging to that class.

4.2 Implementation Details

For S3DIS dataset, unlike PointNet that adopts 9-dim representation, each point in our model is only represented by 6-dim vector in order to reduce computation cost, namely, normalized XYZ and RGB. The follow-up experimental results also validate that 6-dim input in our model already performs better than the original 9-dim vector in PointNet. During training, each room is split into overlapped blocks of size 1.5 m $\times$ 1.5 m along horizontal directions without height constraint, each containing 6400 points. No overlapping is performed during

Table 1. Comparison results on the S3DIS dataset with XYZ-RGB inputs. IoU data of the referenced works are from [23] and [27]. The upper results are averaged over the 6 folds and the lower are trained on two buildings and tested on the Area 5 fold. Intersection over union of each class are given.

Method	OA	mAcc	mIoU	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Bookca.	Board	Clutter
A5 PointNet [4]	-	49.0	41.1	88.8	97.3	69.8	0.05	3.92	49.3	10.76	58.9	52.6	5.8	40.3	26.3	33.22
A5 SEGCloud[14]	-	57.4	48.9	90.1	96.0	69.9	0	**18.4**	38.3	23.1	70.4	75.9	**40.9**	58.4	13.0	41.6
A5 SPG [27]	85.1	61.7	**54.7**	91.5	97.9	75.9	0	14.2	51.3	**52.3**	77.4	**86.4**	40.4	**65.5**	7.23	50.7
A5 Ours	**85.7**	**71.3**	53.4	**95.2**	**98.6**	**77.4**	**0.80**	9.83	**52.7**	27.9	**78.3**	76.8	27.4	58.6	**39.1**	**51.0**
PointNet [4]	78.5	66.2	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
MS+CU [23]	79.2	59.7	47.8	88.6	**95.8**	67.3	36.9	24.9	48.6	52.3	51.9	45.1	10.6	36.8	24.7	37.5
G+RCU [23]	81.1	66.4	49.7	90.3	92.1	67.9	**44.7**	24.2	**52.3**	51.2	58.1	47.4	6.9	39.0	30.0	41.9
SPG [27]	82.9	64.4	54.1	92.2	95.0	71.9	33.5	15.0	46.5	**60.9**	**69.4**	65.0	**38.2**	56.8	6.86	51.3
Ours	**86.9**	**73.6**	**56.3**	**92.9**	93.8	**73.1**	42.5	**25.9**	47.6	59.2	60.4	**66.7**	24.8	**57.0**	**36.7**	**51.6**

test. The parameters of the proposed pointwise pyramid pooling module are set as follows: 4 pooling layers with the corresponding window kernel size 1, $N/64$, $N/8$, N. With regard to the two-direction hierarchical RNN model, we set time steps to 6 for each direction. The hidden unit size for each RNN cell is 128. For ScanNet [7] we carry out experiments utilizing only the geometry information by removing RGB features. Besides, weighted cross entropy loss is harnessed to tackle with the challenge of imbalanced samples between different classes. To keep fair comparison with PointNet++ [22], we also perform random input dropout for all approaches in the ScanNet experiment. And for vKITTI dataset [10], experiments with XYZ-RGB and with XYZ-only inputs are both conducted. For all experiments, the model is optimized by Adam optimizer [36] with initial learning rate 0.001 and batch size 24.

4.3 Results on Indoor Datasets

S3DIS: Similar to [4, 27, 23] which adopted 6-fold cross validation strategy for train and test, we trained six models separately and for each one, five areas are used for training and the remaining one for testing. Note that SEGCloud [14] trained their model on two of the three buildings and tested on the other building, since they argued that areas from the same building in training and testing set can result in an increased performance and fail to evaluate the generalizability. For fair comparison, we also retrain our model on the first two buildings and test on the building not present in the other folds, namely, Area 5 (A5). The comparison results are presented in Table 1.

As is depicted in Table 1, our architecture outperforms other approaches on average. The proposed method outperforms baseline work PointNet by 8.7% mIoU, 7.4% mAcc and 8.4% in overall accuracy and even shows higher accuracy than [23] and [27], which exploit multi-scale context consolidation and superpoint graph, respectively. Besides, our architecture is able to resolve small semantic classes such as beam, column and board. With regard to the generalizability evaluation on Area 5, the model trained on two buildings and tested

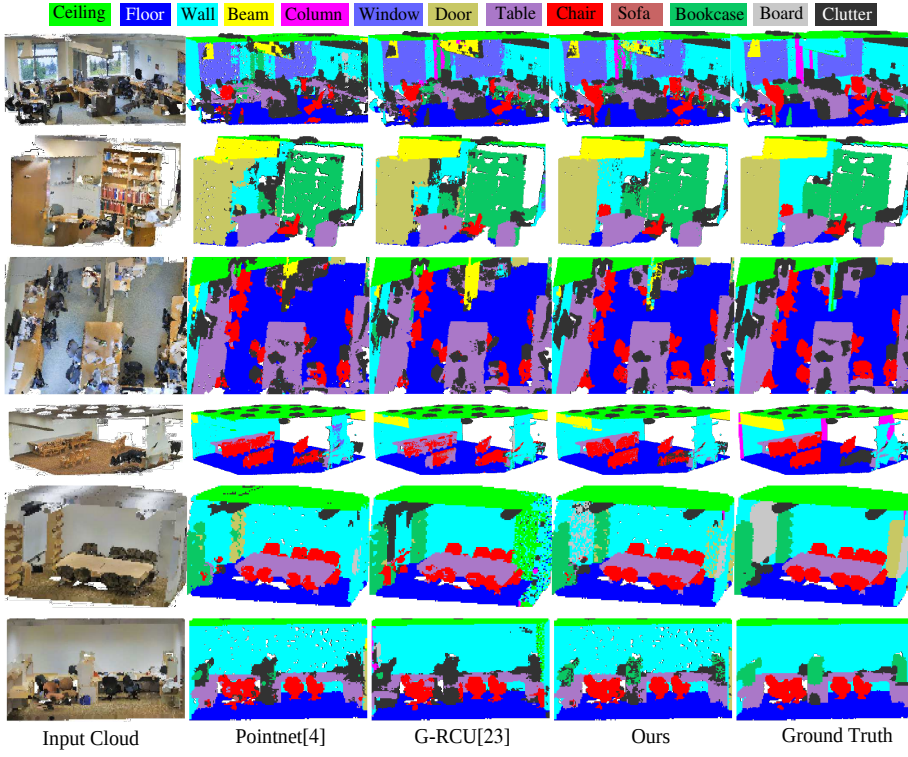

Fig. 4. Qualitative results on indoor S3DIS dataset. Our results demonstrate superior performance over the state-of-the-art methods with more accurate predictions.

on another diverse building also behaves well, leading the performance of overall accuracy and mAcc and yielding comparable IoU with [27].

Next, we present the qualitative results of the semantic segmentation by our architecture in Fig. 4. As can be seen from Fig. 4, thanks to the pyramid pooling module as well as the two-direction hierarchical RNN model, our architecture is capable of correcting erroneously labelled classes in [4, 23] and achieves more accurate segmentation results. Besides, the proposed framework largely retrieves fine-grained details that are missed by other methods. For example, the chair legs are preserved to a great extent (colored in red) and much less noise in semantic segmentation is observed compared to the remaining approaches.

In the previous experiment (S3DIS), geometry as well as color information is utilized to predict semantic labels for each room, since color plays a critical role in feature representation. We wonder whether the proposed architecture works when color is unavailable. Accordingly, a further experiment is conducted on large scanned indoor dataset ScanNet [7] with color information discarded. ScanNet presented a voxel-based coarse prediction framework leveraging 3D fully convolutional networks. Instead, we yield per-point labels and make comparison

Table 2. Per-point accuracy on ScanNet [7] with only XYZ information, no RGB.

Method	PointNet [4]	G+RCU [23]	PointNet++ [22]	Ours
OverAll Accuracy	0.526	0.634	0.743	**0.765**

Table 3. Results on outdoor vKITTI dataset: with and without RGB.

Method	XYZRGB			XYZ-only		
	OA	mIoU	mAcc	OA	mIoU	mAcc
PointNet [4]	0.797	0.344	0.470	0.717	0.239	0.381
G+RCU [23]	0.806	0.362	0.497	0.739	0.298	0.467
PointNet++ [22]	-	-	-	0.770	0.299	0.400
Ours	**0.878**	**0.416**	**0.541**	**0.796**	**0.345**	**0.492**

with PointNet [4] and Pointnet++ [22]. The performance is reported in Table 2, which demonstrates the efficiency of our framework when only geometry information is available. Note that the result is a bit different from that in [22], since the accuracy in our experiment is evaluated by per-point rather than per-voxel.

4.4 Results on Outdoor Datasets

We also evaluate the performance of the proposed model on outdoor datasets. For fair comparison, we choose vKITTI dataset [10] as [23] does and split the five different urban video sequences into six non-overlapping folds. During training and testing, the six-fold cross validation strategy is adopted as PointNet suggests. Furthermore, we conduct two separate experiments with respect to different input features, i.e., XYZ-RGB and XYZ-only. As is presented in Table 3, our framework successfully predicts the semantic labels of outdoor scenes no matter which input feature strategy is adopted. With color information, our architecture can improve the semantic segmentation performance to a great extent. Even without color clue, our algorithm is able to achieve improvements compared to other state-of-the-art approaches. Notably we obtain slightly higher averaged performance of [4, 23] reported in [23] using the same dataset, which is probably due to our data normalization. Besides, we also show qualitative results of vKITTI compared to other recently proposed state-of-the-art algorithms in Fig. 5. As is demonstrated in Fig. 5, our framework retrieves the scene more consistently with less erroneous labels.

To further validate the effectiveness and generalization ability of our model, we apply the model trained on vKITTI to two untrained real-world outdoor datasets, KITTI Raw [9] and 3DRMS [35] laser data. The KITTI Raw point clouds obtained by LiDAR scans only contain XYZ information without ground truth semantic labels, thus we apply geometry-only model of vKITTI to it. With regard to the latter 3DRMS laser data with color information, the geometry-color model of vKITTI is employed, despite of different class labels for these two datasets. The qualitative results are presented in Fig. 6. Without any training,

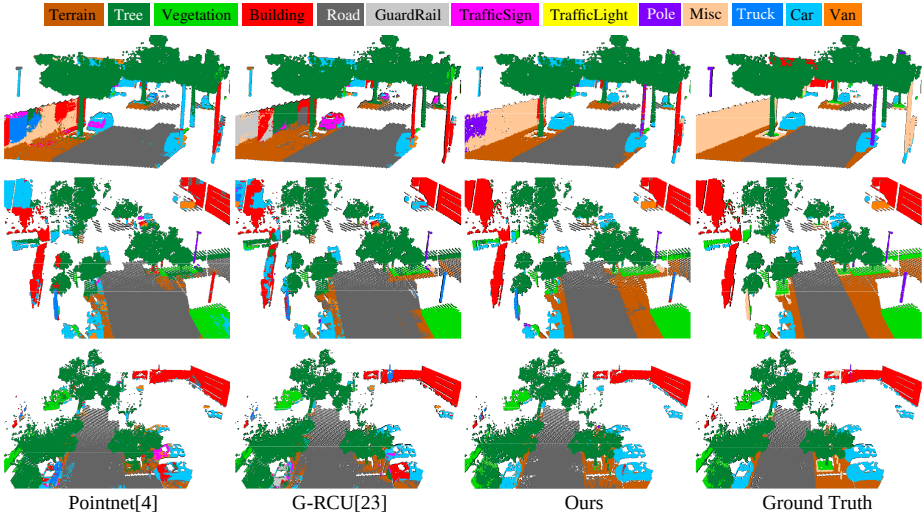

Fig. 5. Semantic segmentation results on outdoor vKITTI dataset. From left to right: PointNet [4], G+RCU [23], our results, the ground truth. The input features are geometry with color for all algorithms.

our model still produces reasonable semantic results in both datasets. Note that only some common or similar classes make sense, such as road, tree, terrain, car and building.

4.5 Ablation Study

For ablation study, further experiments are carried out to explore the contribution of two key components in our approach. For all experiments here, we compare the performance of different settings with geometry and color features taken as input on S3DIS dataset. As is revealed in Table 4, though being simple, the pyramid pooling module makes significant contribution to the improvement of the overall accuracy, and our two-direction RNN model further reduces errors in small classes and thus improves the mIoU and mAcc. Although the results of 1D RNN are inferior to that with pointwise pyramid pooling, the hierarchical

Table 4. Comparison of different variants. Best results are shown in bold.

Method	OA	mAcc	mIoU
Baseline PointNet [4]	78.5	66.2	47.6
Add with Pointwise Pyramid Pooling	82.8	68.3	50.8
Add with one-direction RNNs	80.6	67.9	49.9
Add with two-direction RNNs	82.3	70.0	51.4
Our Full Approach	**86.9**	**73.6**	**56.3**

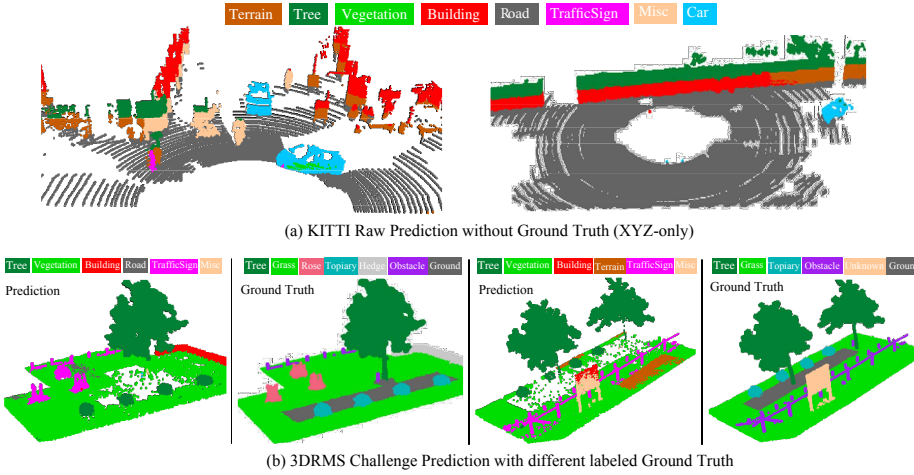

Fig. 6. Qualitative prediction results on the untrained KITTI Raw [9] (upper row) and 3DRMS [35] (below row). Our XYZ-only model trained on vKITTI is applied to real KITTI laser scans. Only some shared classes like car, building and road make sense. The XYZ-RGB model trained on vKITTI is employed for 3DRMS scans. Although some classes differ from the annotated labels, reasonable results are still observed.

Table 5. Results of different time steps on S3DIS dataset (6-fold cross validation).

Time step	OA	mAcc	mIoU
1	80.7	69.8	51.3
2	83.8	72.7	53.7
4	85.3	73.2	**56.4**
6	**86.9**	**73.4**	56.2
8	**86.9**	73.3	55.6
10	86.5	72.1	54.0

two-direction RNN architecture reveals an improved performance. Finally, the combination of the two components achieves overwhelming results by integrating neighboring local context with long-distance spatial information.

Besides, since the best time step varies for different datasets and depends on the trade-off between time and accuracy, thus we didn't make much effort on finetuning the best one. However, we conducted experiments concerning different time steps. Results shown in Table 5 reveal that small time steps degrade the performance whereas too large time step also hinders the IoU, generally, a time step between 4 and 8 is feasible.

5 Conclusions

We present an end-to-end approach for efficient 3D semantic segmentation by means of integrating convolution neural networks with recurrent neural networks. The framework consists of two indispensable components, the pointwise pyramid pooling module with no strides to integrate multi-scale local context and the two-direction hierarchical RNNs to learn long-range spatial dependency. Our architecture successfully improves the accuracy of 3D semantic segmentation on indoor and outdoor datatsets. With regard to some similar semantic classes, our model also has limited capability to distinguish them, such as door and wall. For future work, we plan to investigate on the problem and extend our method to more applications on unstructured point clouds.

References

1. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In: CVPR. (2017) 1–1
2. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015) 3431–3440
3. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Rodríguez, J.G.: A review on deep learning techniques applied to semantic segmentation. CoRR **abs/1704.06857** (2017)
4. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. (2017) 652–660
5. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. (2012) 746–760
6. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: CVPR. (2016) 1534–1543
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. (2017)
8. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In: ISPRS. Volume IV-1-W1. (2017) 91–98
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11) (2013) 1231–1237
10. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtualworlds as proxy for multi-object tracking analysis. In: CVPR. (2016) 4340–4349
11. Huang, J., Suya, Y.: Point cloud labeling using 3d convolutional neural network. In: ICPR. (2016)
12. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. (2015) 922–928
13. Riegler, G., Ulusoy, A.O., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: CVPR. (2016) 3577–3586
14. Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J., Savarese, S.: Segcloud: Semantic segmentation of 3d point clouds. CoRR **abs/1710.07563** (2017)
15. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. CoRR **abs/1612.01105** (2016)

16. Li, Z., Gan, Y., Liang, X., Yu, Y., Cheng, H., Lin, L.: Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In: ECCV. (2016) 541–557
17. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations. (2015)
18. Su, H., Maji, S., Kalogerakis, E., Learnedmiller, E.G.: Multi-view convolutional neural networks for 3d shape recognition. In: international conference on computer vision. (2015) 945–953
19. Qi, C.R., Su, H., Niebner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: CVPR. (2016) 5648–5656
20. Boulch, A., Saux, B.L., Audebert, N.: Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks. In: Eurographics Workshop on 3D Object Retrieval. (2017)
21. Guerry, J., Boulch, A., Le Saux, B., Moras, J., Plyer, A., Filliat, D.: Snapnet-r: Consistent 3d multi-view semantic labeling for robotics. In: ICCV. (2017)
22. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NIPS. (2017) 5099–5108
23. Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B.: Exploring spatial context for 3d semantic segmentation of point clouds. In: ICCV. (2017)
24. Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J.: PCPNET: learning local shape properties from raw point clouds. CoRR **abs/1710.04954** (2017)
25. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from RGB-D data. CoRR **abs/1711.08488** (2017)
26. Qi, X., Liao, R., Jia, J., Fidler, S., Urtasun, R.: 3d graph neural networks for rgbd semantic segmentation. In: ICCV. (2017)
27. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. CoRR **abs/1711.09869** (2017)
28. Park, H., Lee, K.M.: Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters (2016)
29. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with lstm recurrent neural networks. In: CVPR). (2015) 3547–3555
30. Stollenga, M.F., Byeon, W., Liwicki, M., Schmidhuber, J.: Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In: NIPS. (2015) 2998–3006
31. Pinheiro, P.H.O., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: ICML. (2014) 82–90
32. Visin, F., Romero, A., Cho, K., Matteucci, M., Ciccone, M., Kastner, K., Bengio, Y., Courville, A.C.: Reseg: A recurrent neural network-based model for semantic segmentation. In: CVPR. (2016) 426–433
33. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8) (1997) 1735–1780
34. Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. empirical methods in natural language processing (2014) 1724–1734
35. Torsten, S., Thomas, B., Marc, P., Robert, F., Radim, T.: 3d reconstruction meets semantics – reconstruction challenge. `http://trimbot2020.webhosting.rug.nl/events/3drms/challenge/` (2017) ICCV Workshops.
36. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: ICLR. (2015)