Attention-Based Joint Semantic-Instance Segmentation of 3D Point Clouds

Wen HAO[1,2], Hongxiao WANG[1,2], Wei LIANG[1,2], Minghua ZHAO[1,2], Zhaolin XIAO[1,2]
[1]*School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China*
[2]*Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an, China*
haowen@xaut.edu.cn

***Abstract*—In this paper, we propose an attention-based instance and semantic segmentation joint approach, termed ABJNet, for addressing the instance and semantic segmentation of 3D point clouds simultaneously. First, a point feature enrichment (PFE) module is used to enrich the segmentation network's input data by indicating the relative importance of each point's neighbors. Then, a more efficient attention pooling operation is designed to establish a novel module for extracting point cloud features. Finally, an efficient attention-based joint segmentation module (ABJS) is proposed for combining semantic features and instance features in order to improve both segmentation tasks. We evaluate the proposed attention-based joint semantic-instance segmentation neural network (ABJNet) on a variety of indoor scene datasets, including S3DIS and ScanNet V2. Experimental results demonstrate that our method outperforms the start-of-the-art method in 3D instance segmentation and significantly outperforms it in 3D semantic segmentation.**

***Index Terms*—computer graphics, object segmentation, feature extraction, pattern recognition, machine learning.**

I. INTRODUCTION

Recent growth in autonomous driving and robotics applications has increased demand for 3D scene understanding and perception [1][2]. Semantic segmentation and instance segmentation of 3D scenes are critical components of 3D scene understanding. Semantic segmentation divides the 3D scene into informative regions and assigns each region to a specific class. Instance segmentation classifies objects at the point level in a scene and also distinguishes between instances belonging to the same semantic category. Both tasks share some common ground that can be leveraged associatively to improve their performance. Semantic segmentation is required to categorize point clouds, which is one of the objectives of instance segmentation. Consistent with semantic segmentation, instance segmentation assigns the same label to points that belong to the same instance.

Motivated by the correlation between semantic segmentation and instance segmentation, ASIS [3] firstly proposed a mutual aid module to enable these two tasks to benefit from each other. ASIS adopts *k* nearest neighbor (*KNN*) search to find out *k* nearest neighbors for the center point in instance embedding space, then fuses the semantic features of *k* neighbors to the center point. However, the *KNN* approach treats all semantic information about each neighbor equally, ignoring critical features. We propose an attention-based approach that compensates for this shortcoming by fusing semantic information using learnable soft weights. It adopts a sophisticated attention mechanism to automatically learn critical local features.

This paper introduces the attention-based joint semantic-instance segmentation neural network (ABJNet), which is used to model the interaction between semantic and instance segmentation for the purpose of jointly addressing them. The proposed network ABJNet is composed of four components: a point feature enrichment (PFE) module, a shared feature encoder with attentive pooling operation, two paralleled branch decoders, and an attention-based joint segmentation (ABJS) module.

To learn more effective high-level semantic features, the feature encoder and decoder are built based on PointNet++ [4]. To fully exploit contextual information contained in point clouds, a PFE module is used to capture contextual attention features for each point by indicating the relative importance of its neighbors. To further improve segmentation performance in our ABJNet, we propose a novel attentive-based joint instance and semantic segmentation module that promotes instance and semantic segmentation mutually.

In summary, the main contributions of our work are as follows:

(1) We propose a set abstract module with attentive pooling (AP) operation to identify the most important local features. The powerful attention mechanism is used to determine the most important neighboring point features.

(2) We develop a novel efficient ABJS module to exploit the potential reciprocal information in semantic and instance segmentation tasks to seamlessly fuse the heterogeneous features, allowing these two tasks to benefit from each other. This module would benefit from the discriminative feature by taking advantage of instance and semantic segmentation.

(3) Experiments demonstrate that our ABJNet outperforms the state-of-the-art methods in both semantic and instance segmentation criteria on the S3DIS and ScanNet V2 datasets.

The remainder of the paper is organized as follows. Section II provides an overview of instance segmentation. Section III presents the architecture overview of instance segmentation. The details of our ABJNet are proposed in Section IV. Experimental results are presented in section V. The last section discusses the method's limitations and makes recommendations for future research.

This work was supported in part by National Natural Science Foundation of China under Grant No. 61602373; and Shaanxi Natural Science Foundation under Grant No. 2021JM-342, 2019JQ-740; and the Key Laboratory Research project of Shaanxi Provincial Education Department under grant No.18JS078.

Digital Object Identifier 10.4316/AECE.2022.02003

II. Related Work

Efficient instance and semantic segmentation of point clouds are of great significance in computer vision. With the rapid development of deep learning, many efficient and powerful deep learning network architectures have been proposed for performing semantic segmentation by directly processing points [5-7]. Subsequently, numerous excellent models for instance segmentation were proposed. The instance segmentation method in point clouds is further developed based on semantic segmentation. It then segments each object belonging to the same category and generates an instance label for each point. Existing methods for instance segmentation can be classified as proposal-based, proposal-free, and semantic-instance segmentation fused methods.

(1) Proposal-based instance segmentation

The proposal-based method requires the acquisition of region proposals and further predicts each instance by progressively calculating and refining the region proposal.

Hou [8] proposed a 3D Semantic Instance Segmentation (3D-SIS) neural network architecture for 3D semantic instance segmentation in commodity RGB-D scans. This method jointly learns from both geometric and color signals, thus enabling accurate instance predictions. Yi [9] proposed an instance segmentation network structure named Generative Shape Proposal Network (GSPN) to generate 3D proposals of objects. The GSPN is integrated with the Region-based PointNet (R-PointNet), which enables flexible proposal refinement and instance segmentation. The final label is determined by predicting the binary mask of each class label point-by-point. Yang [10] proposed 3D-BoNet for point cloud segmentation. This method generates rough 3D bounding boxes for all possible instances directly and then labels them using a point-level binary classifier. Zhang [11] proposed the use of self-attention blocks to aid in the learning of feature representations from a bird's-eye perspective. Final instance labels are determined based on the predicted horizontal center and height constraints.

While proposal-based methods are intuitive, they always require multi-stage training and make it difficult to delineate the object instance using a bounding box regression.

(2) Proposal-free instance segmentation

The proposal-free instance segmentation method regards instance segmentation as a post-processing task for semantic segmentation. They predict instances as clusters in feature space using a clustering algorithm.

SGPN [12] is the pioneering work that uses deep learning to process 3D instance segmentation tasks. This method first uses PointNet [13] or PointNet++ [4] to extract a descriptive feature vector for each point. Then a similarity matrix is introduced to indicate the degree of similarity between each pair of points in embedded feature space. The similarity between the two hinge losses is used to adjust the similarity matrix and the semantic segmentation result, allowing for the learning of more discriminative features. Finally, the non-maximum suppression method is used to fuse similarities into examples. Similarly, Liu [14] first leveraged submanifold sparse convolution [15] to predict the semantic scores of each voxel and their affinity with neighboring voxels. They then introduced a clustering algorithm that used multi-scale affinity fields and semantic prediction to group points into instances. Liang [16] proposed a structure-aware loss function that considers both structural and embedding information. An attention-based graph convolutional network is introduced to further refine the learned features adaptively by correlating information from different neighboring points. The mean-shift algorithm [17] is used to cluster refined embeddings to obtain the final instance predictions. PointGroup [18] proposed a clustering algorithm and applied it on the original point set as well as offset-shifted point coordinate set to generate some instance candidates. He [19] proposed a dynamic convolution-based framework called DyCo3D for 3D instance segmentation.

Proposal-free methods grouped the objectness of instance segments is usually low since these methods do not explicitly detect object boundaries [20].

(3) Semantic-instance segmentation fused method

The previous works have tended to tackle the two tasks independently, without exploring the underlying relationship between them. Since there is some independence between semantic and instance segmentation tasks, many researchers combine the two tasks into a single one by associatively segmenting semantics and instances. Semantic and instance segmentation can complement one another in the semantic-instance segmentation fused method.

Wang [3] proposed an associatively segmenting instances and semantics (ASIS) module to closely associate instance segmentation and semantic segmentation. ASIS enables semantic and instance segmentation to take advantage of each other, resulting in a win-win situation. Pham [21] proposed a multi-task pointwise network (MT-PNet), for classifying 3D points and embedding them in a high-dimensional features space. Then, the multi-value conditional random field (CRF) model is used to handle joint segmentation tasks by integrating both 3D and high dimensional embedded features. Similarly, Zhao [22] proposed JSNet to simultaneously address the instance and semantic segmentation of 3D point clouds. The framework is composed of a shared feature encoder, two parallel feature decoders, and a point cloud feature fusion (PCFF) module, as well as a joint instance semantic segmentation (JISS) module. On instance embeddings, simple mean-shift clustering is used to generate instance predictions. Wu [23] developed a Bi-Directional Attention module for 3D point cloud perception based on backbone neural networks. This module utilizes a similarity matrix calculated from the features of one task to help aggregate non-local information for the other task, avoiding feature exclusion and task conflict. JSPNet [24] developed a feature fusion module based on similarity that locates the inconspicuous area in the current branch's feature and then selects related features from the other branch to compensate for the unclear content. To establish the probabilistic correlation between semantic and instance features, a cross-task probability-based feature fusion module is developed.

The primary disadvantage of the semantic-instance segmentation fused method is that it typically treats each neighbor's semantic information equally, which always ignores important features.

III. Overview

The entire network is depicted in Figure 1, which includes a PFE module, a shared feature encoder, two parallel

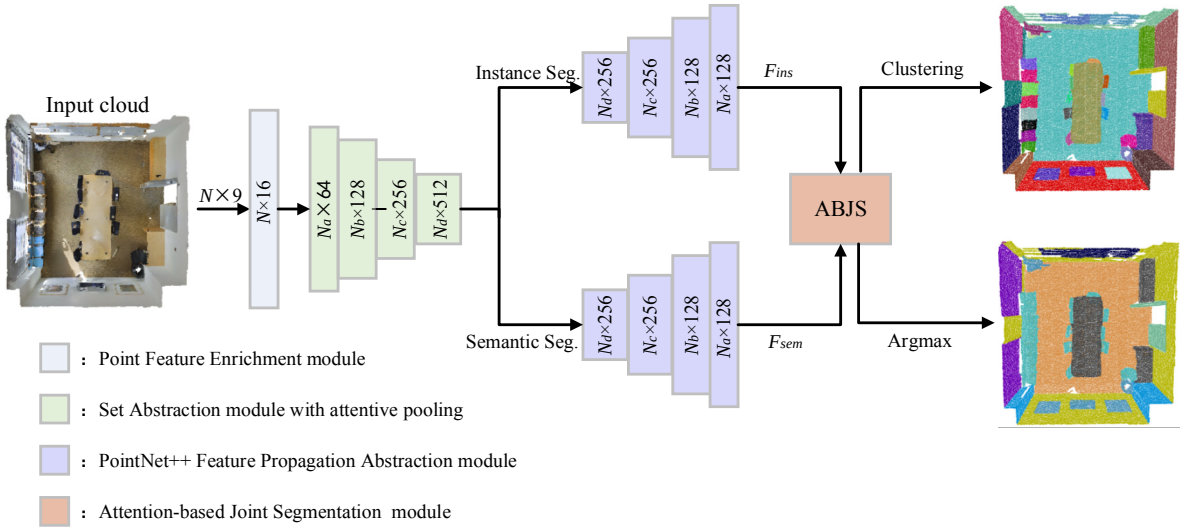


Figure 1. Architecture of our network

decoders, and a module for each decoder that performs attention-based feature fusion. One task is to extract semantic features for each point, and the other is to perform instance segmentation. We can directly use PointNet++ as our backbone network for the two decoders, as the two decoders have the same structure.

The input of our network is *Nx9* for the whole pipeline. *N* is the number of point cloud represented by a 9-dim vector (XYZ, RGB and normalized location as to the room). For Both S3DIS and ScanNet V2 datasets, each room are split into 1m×1m overlapped blocks, each containing 4096 points.

First, a PFE module is defined to enhance the representation of point features. After constructing a graph of the point's *k*-nearest neighbors, the contextual attention feature for each point is computed by embedding the graph attention mechanism within stacked Multi-Layer-Perceptron (MLP) layers.

The point's coordinates and contextual attention feature are then passed to the encoder module. The first layer of the encoder then introduces the set abstraction of PointNet++ into the model. The pooling layer realizes the abstraction of the point set.

Due to the possibility of losing detailed information when using max-pooling, attentive pooling is used in the grouping layers. Following that, the output of the feature encoder is fed into two parallel decoders and processed separately by their subsequent components. Finally, an ABJS module is proposed, which consists of two coupled instance-to-semantic and semantic-to-instance streams for extracting useful information while filtering out useless information.

The semantic segmentation branch is supervised by the classic cross-entropy loss at training time. In terms of instance segmentation, we use the agnostic loss in [3] to supervise instance embedding learning, which draws points belonging to the same instance object together and maintains a greater distance between points belonging to different instances. Final instance labels can be obtained at test time by applying the mean-shift clustering [17] algorithm to the instance embedding.

IV. Proposed Method

A. Encoder

(1) Point feature enrichment (PFE) module

Extraction of local geometric features is critical for accurate segmentation. However, context features between points can be quite useful for segmenting point clouds. As a result, we use a PFE module as a preprocessor for the raw data.

The PFE module [25] is proposed to capture contextual attention features by indicating different importance of each point's neighbors. The PFE module simultaneously learns self-attention and neighboring-attention features and then fuses them via a non-linear activation function leaky RELU to obtain attention coefficients. Additionally, they are normalized using a softmax function. Then, a linear combination operation is applied to finally generate the attention feature. The self-attention mechanism learns self-coefficients by considering the self-geometric information associated with each point, whereas the neighboring-attention mechanism concentrates on local coefficients by considering the neighborhood.

(2) Set abstraction layer

Following the PFE module, the first layer of the encoder incorporates the PointNet++ set abstraction into our model. The sampling and grouping layers in PointNet++ were used to obtain the local structure. The max-pooling operation was used to aggregate the encoded local information in the local region, which helps reduce the dimensionality of the features while also filtering out unreliable noise. However, the max-pooling operation may result in the loss of some useful information. As a result, we introduce an attention pooling [26] operation over the neighborhood in our work to identify the most critical features to further obtain local signature representation and enhance the robustness of the network.

Given the set of local features $F=\{f_1, f_2, \ldots, f_n\}$, we use a shared function g(·) to learn a unique attention score for each feature. The function g(·) is composed of a shared MLP and an activation function of softmax, which is defined as follows:

$$S_i = g(f_i, W) \tag{1}$$

where, W is the learnable weights of a shared MLP. The final feature vector F_W is the weighted summer as follows:

$$F_W = \sum_{i=1}^{n} (f_i \cdot S_i) \tag{2}$$

B. Mutual Aid

(1) Feature attentive aggregation module

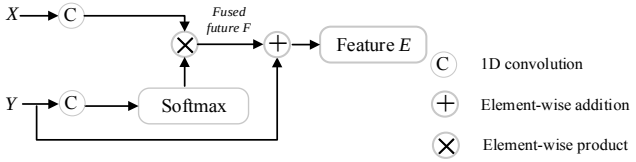


Figure 2. The FAA module

We propose a feature attentive aggregation module (FAA) for fusing two features. Figure 2 illustrates the details of the structure. The FAA module has two inputs: feature X and feature Y.

First, features X and Y are transformed through a 1×1 convolution, respectively. The attentive scores of transforming feature Y are computed by a softmax operation. Then an element-wise multiply operation of feature X and attentive scores are applied to generate the fused feature F. Finally, the fused feature F and the original instance feature Y perform an element-wise sum operation to produce the final feature E:

$$E = F + Y \tag{3}$$

As can be inferred from Eq. (3), the final feature at each position is a sum of the fused features at all positions and original instance features. FAA module can balance two features and refine the output features from the decoder with better feature representation.

(2) Attention-based joint segmentation (ABJS) module

We propose an ABJS module to obtain semantic labels and segment instance objects simultaneously, allowing semantic and instance segmentation tasks to benefit from each other. This module is applied to fuse semantic and instance features.

Figure 3 illustrates the details of the ABJS module. F_{ins} and F_{sem} denote the output feature matrices of the two parallel decoder branches, respectively. For the instance segmentation task, the semantic feature matrix F_{sem} is transformed into instance feature space as F_{si} by a 1D convolution. The F_{si} is added to the instance feature matrix F_{ins} element-wise as F_{sis}. Furthermore, the semantic-aware instance feature F_{sis} and semantic feature matrix F_{sem} are fed into the FAA module, with the semantic feature guiding the instance segmentation task. Following the FAA module, the output feature is combined with the original instance feature F_{ins} to obtain the enhanced feature F_{sisa}.

Finally, three 1D convolutions are performed in the enhanced feature F_{sisa} to generate the instance embedding feature E_{ins}. The preceding procedure can be formulated as follows:

$$F_{sis} = F_{ins} + Conv1D(F_{sem}) \tag{4}$$

$$F_{sisa} = FAA(F_{sis}, F_{sem}) + F_{ins} \tag{5}$$

$$E_{insm} = Conv1D(Conv1D(F_{sisa})) \tag{6}$$

$$E_{ins} = Conv1D(E_{insm}) \tag{7}$$

The final instance labels are generated after performing mean-shift clustering on E_{ins}.

For the semantic segmentation task, the middle feature E_{insm} obtained by two 1D convolutions in the instance segmentation branch is added to the semantic feature matrix F_{sem} element-wise as F_{sim}.

The instance-fused semantic feature F_{sim} and instance feature matrix F_{ins} are then passed to the FAA module, which refines the semantic segmentation task using the instance.

The output feature of the FAA module was combined with the original semantic feature F_{sem} to obtain the fused feature F_{sima}. Three 1D convolutions are performed on F_{sima} subsequently to obtain the semantic feature P_{sem}, which is used to predict the semantic categories. We also formulate this procedure as follows:

$$F_{sim} = F_{sem} + E_{insm} \tag{8}$$

$$F_{sima} = FAA(F_{sim}, F_{ins}) + F_{sem} \tag{9}$$

$$P_{sem} = Conv1D(Conv1D(Conv1D(F_{sima}))) \tag{10}$$

V. Experimental Results

A. Datasets and Evaluation Metrics

We evaluate our approach on two public datasets: Stanford 3D Indoor Semantics Dataset (S3DIS) [27] and Richly-annotated 3D Reconstructions of Indoor Scenes ScanNet V2 [28]. S3DIS is an indoor 3D point cloud dataset consisting of 3D scans of Matterport scanners from 6 areas. There are 271 rooms divided by room. Each point in the scene is associated with an instance annotation and one of the semantic labels from 13 categories. For S3DIS dataset, each point has a 9-dimensional feature vector including XYZ, RGB, and normalized coordinates. Following PointNet, we split the rooms into 1 m × 1 m overlapping blocks with a stride of 0.5m on the ground plane. Each block contains 4096 points in total. ScanNet V2 is an RGB-D video dataset containing 1513 scans with 3D object instance annotations. We adopt the same strategy as S3DIS that rooms are split into 1 m × 1 m overlapping blocks with a stride of 0.5 m and sample 4096 points from each block.

(1) Evaluation metrics

For evaluation of semantic segmentation, overall accuracy (oAcc), mean accuracy (mAcc), and mean Intersection over Union (mIoU) is calculated across all the categories. For 3D instance segmentation, mean precision (mPrec), mean recall (mRec) with 0.5 IoU threshold, coverage (Cov) and weighted coverage (WCov) are adopted to evaluate our method. Cov denotes the average instance-wise IoU of prediction matched with ground-truth. The score is further weighted by the size of ground-truth instances to obtain Wcov.

Given the ground-truth regions G and predicted regions O, Cov and $WCov$ are calculated as:

$$Cov(G,O) = \sum_{m=1}^{|G|} \frac{1}{|G|} \max_n IoU(r_m^G, r_n^O) \tag{11}$$

$$WCov(G,O) = \sum_{m=1}^{|G|} \omega_m \max_n IoU(r_m^G, r_n^O) \tag{12}$$

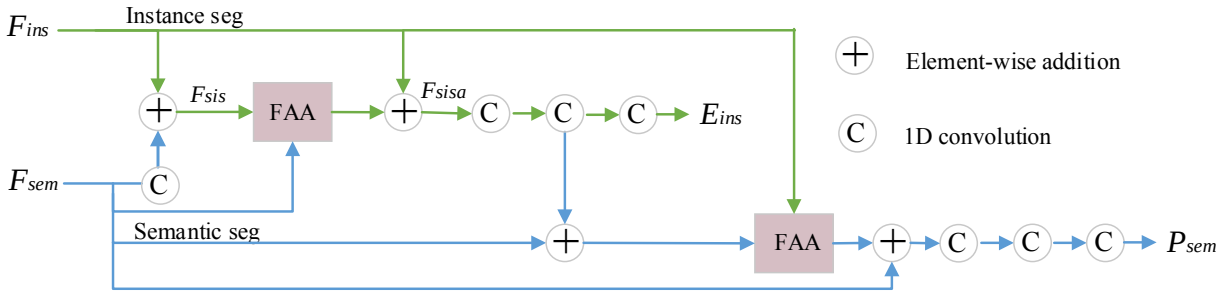


Figure 3. Attention-based Joint Segmentation (ABJS) module

$$\omega_m = \frac{\left|r_m^G\right|}{\sum_k \left|r_k^G\right|} \tag{13}$$

where, $\left|r_m^G\right|$ represents the total number of points in ground truth region *m*.

(2) Implementation details

All the experiments are conducted on a PC with Intel(R) i7-8700 CPU, 32G memory and a single NVIDIA RTX 2080Ti GPU. We train the network for 100 epochs with batch size of 12 and select the Adam optimizer to optimize the network on a single GPU. Momentum is set to 0.9. Base learning rate is set to 0.001, and decays by 2 every 300*k* iterations. During the inference process, we also apply mean-shift clustering with bandwidth 0.6 to generate instance objects. BlockMerging algorithm is applied to merge instances from different blocks.

B. Evaluation and Comparison

In this section, we comprehensively evaluate ABJNet and compare it with the state-of-the-art segmentation methods. We evaluate our model in the following aspects: (1) Area 5 is treated as the testing, while residuals are used for training, and (2) 6-fold cross validation that each area is treated as the testing once.

(1) Quantitative results on the S3DIS dataset

Semantic segmentation

Table I shows the performance of our method on semantic segmentation task on S3DIS. We compare our method with several state-of-the-art semantic segmentation methods including ASIS [3], BAN [23], 3DCFS [29] and ISSF [30]. Our ABJNet achieves 74.8 mAcc in 6-fold cross validation and 65.7 mAcc in Area 5 which is superior to other methods on S3DIS. Compared with ASIS, our model improves the indicators of mAcc, oAcc, and mIoU by 4.7, 2.2, and 4.9 on 6-fold cross validation experiments. When evaluated by Area 5 of S3DIS, the improvements are still significant: 4.8 mAcc, 2.5 oAcc and 5.6 mIoU gains. Note that all the precision units are percents (%). The performance has been improved obviously in all three evaluation metrics.

In addition, we also compare our method with other start-of-the-art methods including BAN, 3DCFS and ISSF. Our method outperforms these methods with a significant margin on 6-fold cross validation and Area 5 of S3DIS.

To intuitively present our results, some visualization of semantic segmentation results are shown in Figure 4. At the same time, qualitative comparison of ASIS, BAN and our method are shown in Figure 4(b), (c) and (d). For semantic segmentation, different colors represent different categories. In the first row, the board in ASIS and BAN are recognized as another category by mistake. In the second row, the wall near the board in ASIS is wrongly recognized as the column and some points of the beam in BAN are wrongly recognized as the ceiling. In the last row, the windows (blue color) in BAN are segmented incompletely. In addition, the door in our method achieves a better segmentation effect which proves the effectiveness of our method. The proposed ABJNet segmented points more accurately with respect to the corresponding categories.

TABLE I. COMPARISON OF SEMANTIC SEGMENTATION RESULTS ON S3DIS DATASET

	Methods	mAcc(%)	oAcc(%)	mIoU(%)
6-fold	ASIS[3]	70.1	86.2	59.3
	BAN[23]	71.7	87.0	60.8
	3DCFS[29]	72.4	86.3	60.3
	ISSF[30]	71.6	86.7	60.9
	Ours	**74.8**	**88.4**	**64.2**

	Methods	mAcc(%)	oAcc(%)	mIoU(%)
Area 5	ASIS[3]	60.9	86.9	53.4
	BAN[23]	62.5	87.7	55.2
	3DCFS[29]	62.7	87.8	55.5
	ISSF[30]	62.7	87.7	55.3
	Ours	**65.7**	**89.4**	**59.0**

Instance segmentation

To validate the performance of our method in instance segmentation, Table II depicts the experimental comparison with state-of-the-art methods on S3DIS. Compared with ASIS, the improvements of our model are significant on four evaluation metrics: 3.1 for mCov, 3.1 for mWCov, 0.8 for mPrec and 2.9 for mRec on 6-fold cross validation experiments. When evaluated on Area 5, our method is also better than BAN and 3DCFS.

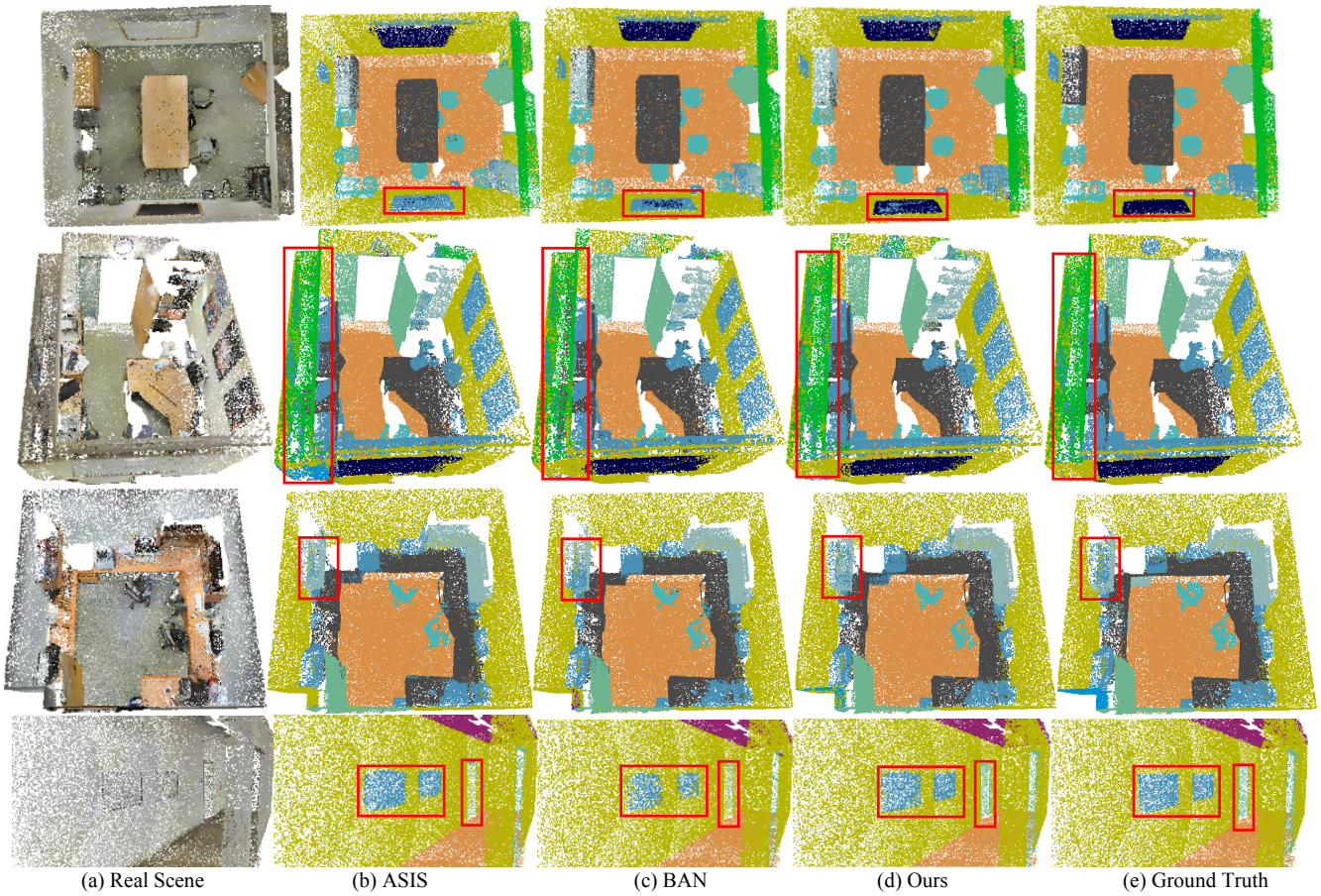
Table III shows the instance and semantic segmentation results for specific categories. Per-class Wcov is shown in the first row. We find our method can outperform ISSF in 9 out of 13 classes. Our method yields large Wcov gains on class “door”, class “bookcase”, and class “board”. Per-class mIoU is shown in the second row. Our method can outperform ISSF in 11 out of 13 classes. The performance has been improved obviously in most classes.

Moreover, qualitative results of instance segmentation are illustrated in Figure 5, which indicates the well-segmented instance capability of our method. For instance segmentation, different instances are represented by different colors. In the first row, the clock on the top right wall is not segmented in ASIS and BAN. In the second row, two windows on the wall are also not separated in ASIS and BAN. In the third row, two adjacent chairs are incorrectly

segmented into one instance in ASIS and BAN. In the last row, three windows on the right wall are also not segmented into a single instance either. The proposed method precisely distinguished between different instances, especially for instances belonging to the same category.

TABLE II. COMPARISON OF INSTANCE SEGMENTATION RESULTS ON S3DIS DATASET

	Methods	mCov (%)	mWCov (%)	mPrec (%)	mRec (%)
6-fold	ASIS[3]	51.2	55.1	63.6	47.5
	BAN[23]	52.1	56.2	63.4	**51.0**
	3DCFS[29]	53.1	57.1	63.7	49.1
	ISSF[30]	54.2	58.1	**65.3**	50.8
	Ours	**54.3**	**58.2**	64.4	50.4
Area 5	ASIS[3]	44.6	47.8	55.3	42.4
	BAN[23]	49.0	52.1	56.7	45.9
	3DCFS[29]	49.0	52.1	55.5	45.9
	ISSF[30]	48.7	51.8	**58.2**	46.6
	Ours	**50.3**	**53.5**	57.8	**48.2**

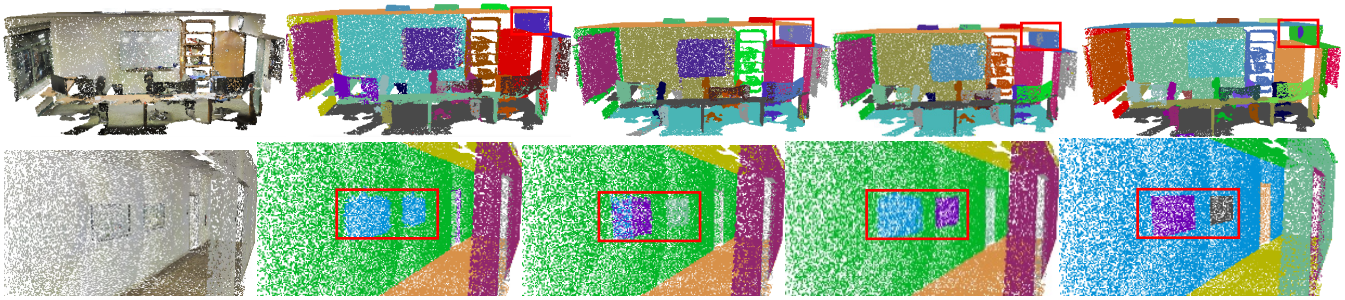


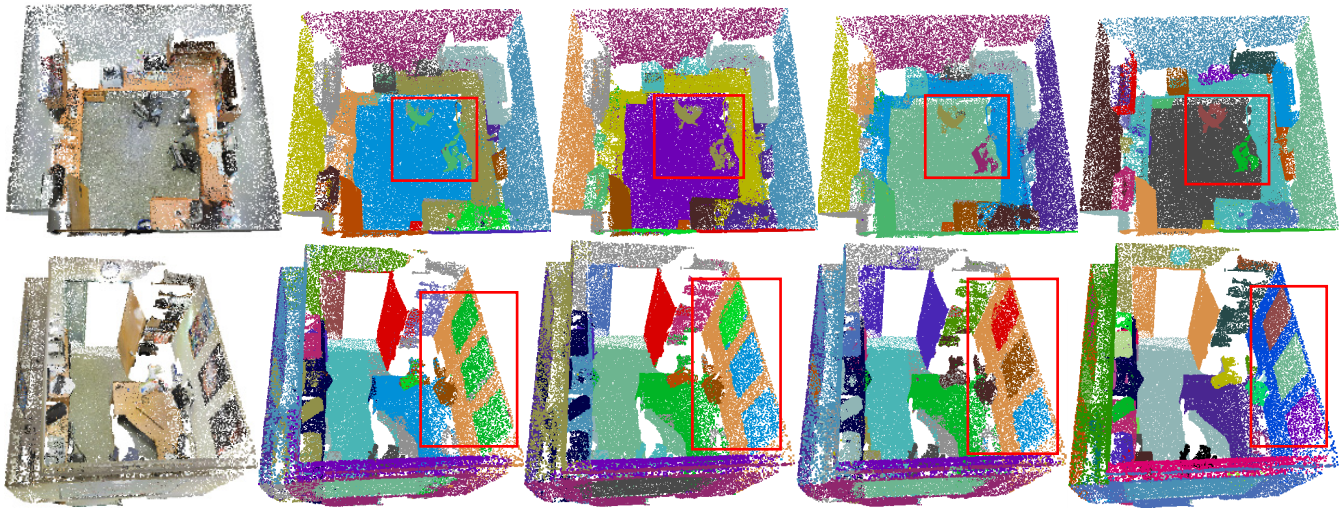
(a) Real Scene (b) ASIS (c) BAN (d) Ours (e) Ground Truth

Figure 4. Comparison result of ASIS, BAN, and our method on semantic segmentation task on S3DIS

TABLE III. PER CLASS RESULTS ON S3DIS DATASET. BOLD MEANS THE BEST METRICS

Metrics	Method	mean	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Wcov	ISSF[30]	58.1	82.8	78.0	70.4	53.9	**22.1**	**64.9**	54.6	58.3	**69.8**	**41.3**	41.3	44.2	51.0
	Ours	**58.2**	**82.9**	**79.3**	**71.4**	**54.1**	16.4	64.4	**60.3**	**59.2**	64.1	41.0	**44.9**	**65.0**	**53.0**
IoU	ISSF[30]	60.9	93.4	94.7	76.4	47.7	**40.8**	**58.4**	62.7	67.7	59.5	31.7	52.4	52.0	53.9
	Ours	**64.2**	**93.8**	**96.6**	**77.7**	**50.4**	34.8	55.6	**67.4**	**71.2**	**71.1**	**42.5**	**56.8**	**56.9**	**58.5**





(a) Real Scene (b) ASIS (c) BAN (d) Ours (e) Ground Truth

Figure 5. Comparison result of ASIS, BAN and our method on instance segmentation task on S3DIS

(2) Quantitative results on the ScanNet V2 dataset

We further make a thorough experiment on ScanNet V2 which is the biggest indoor 3D point cloud dataset by now. We reproduced the results of ASIS [2], JSNet [22] and BAN [23] using the code at GitHub published by the respective authors to make a comparison with the same PointNet++ backbone. The performance comparison of semantic and instance segmentation between ASIS, BAN, JSNet with our method is shown in Table IV and Table V, respectively.

TABLE IV. COMPARISON OF SEMANTIC SEGMENTATION RESULTS ON SCANNET V2 DATASET

Methods	mAcc(%)	oAcc(%)	mIoU(%)
ASIS[3]	48.7	73.0	38.2
JSNet[22]	52.3	73.2	40.3
BAN[23]	49.4	72.8	38.2
Ours	**55.5**	**74.7**	**43.1**

As shown in Table IV, ABJNet achieves 55.5 mAcc, 74.7 oAcc and 43.1 mIoU, which dramatically outperforms ASIS by 6.8 for mAcc, 1.7 for oAcc and 4.9 for mIoU on the semantic segmentation task, respectively. At the same time, the ABJNet outperforms JSNet by 3.2 for mAcc, 1.5 for oAcc and 2.8 for mIoU.

TABLE V. COMPARISON OF INSTANCE SEGMENTATION RESULTS ON SCANNET V2 DATASET

Methods	mCov (%)	mWCov (%)	mPrec (%)	mRec (%)
ASIS[3]	27.8	29.0	33.2	26.1
JSNet[22]	31.4	32.6	**35.9**	**32.0**
BAN[23]	27.6	28.8	30.4	26.3
Ours	**32.6**	**33.9**	**35.9**	**32.0**

As shown in Table V, our ABJNet achieves 32.6 mCov, 33.9 mWCov, 35.9 mPrec and 32.0 mRec which significantly outperforms the state-of-the-art methods ASIS and BAN by a large margin. Our method further outperforms JSNet by 1.2 for mCov and 1.3 for mWCov. The stable improvement in both semantic and instance segmentation demonstrates our novel modules can catch the relationship between semantic and instance features better than the state-of-the-art methods.

ScanNet V2 is the biggest indoor 3D point cloud dataset by now which contains a diverse set of spaces ranging from small to large. Compared with S3DIS dataset, ScanNet V2 dataset has more serious occlusion. It is proper to sample 8192 points from each block to obtain better segmentation results. However, high sampling data cannot run on our computer successfully because of the memory limitation. We split ScanNet V2 dataset into 1 m × 1 m overlapping blocks and sample 4096 points from each block. Therefore, the segmentation results for ScanNet V2 are worse than those for S3DIS.

Furthermore, Figure 6 also illustrates the qualitative semantic visualization of our method on the ScanNet V2 dataset. Each column in Figure 6 represents the segmentation results of ASIS, BAN, JSNet, our method and ground truth, respectively. It can be seen that our method has better segmentation effects. In the first row, the front wall (brown points) is not segmented accurately in ASIS, BAN and JSNet. The sofas in the second row should be segmented into the same class. However, part of sofas (purple points) in ASIS, BAN and JSNet is wrongly classified as another category. In the third row, part of the tables in the room center are also classified by mistake. The proposed method performs better on classifying the entire semantic information.

Figure 7 illustrates the qualitative instance visualization of our method on the ScanNet V2 dataset. Each column in Figure 7 represents the segmentation results of ASIS, BAN, JSNet, our method and ground truth, respectively. In the first row, three chairs around a table are not segmented into a individual instance. In the second row, the bookcases in ASIS, BAN and JSNet are oversegmented. Our results are essentially the same as the ground truth.

To evaluate the computation and memory required of our networks, ASIS and BAN, we report the computation time and memory of training and testing process in Table VI. For a fair comparison, all codes are run in the same environment, including the same GPU (RTX 2080Ti), batch size (12) and data (Area 5 including 68 rooms). Note that all the time units are minutes, and all the memory units are MB. The result is the time and memory required for one epoch in the training and testing process. As we can see, ASIS needs relatively more time for training because ASIS adopts k nearest neighbor search to find out k nearest neighbors for the center point in instance embedding space. The construction of high-order sparse matrices needs to occupy a

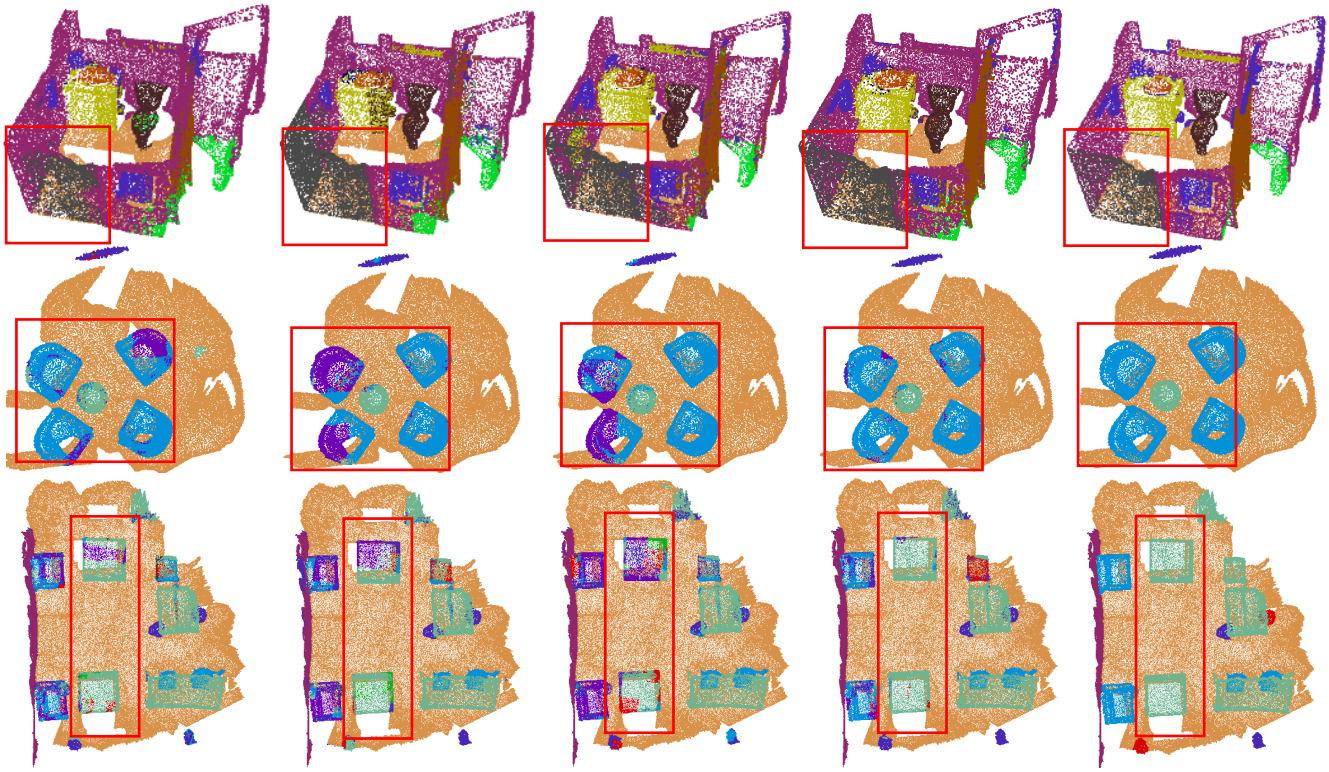
large memory. Our approach needs 9.8 minutes while acquiring better performance, which is faster and more efficient than the state-of-the-art methods.

C. Ablation Study

To further validate the effectiveness of each component proposed in our network, we design ablation experiments and display them on Area 5 of S3DIS dataset. The baseline network includes one shared encoder and two decoders, both of which are built by stacking set abstraction and feature propagation modules from PointNet++, respectively.

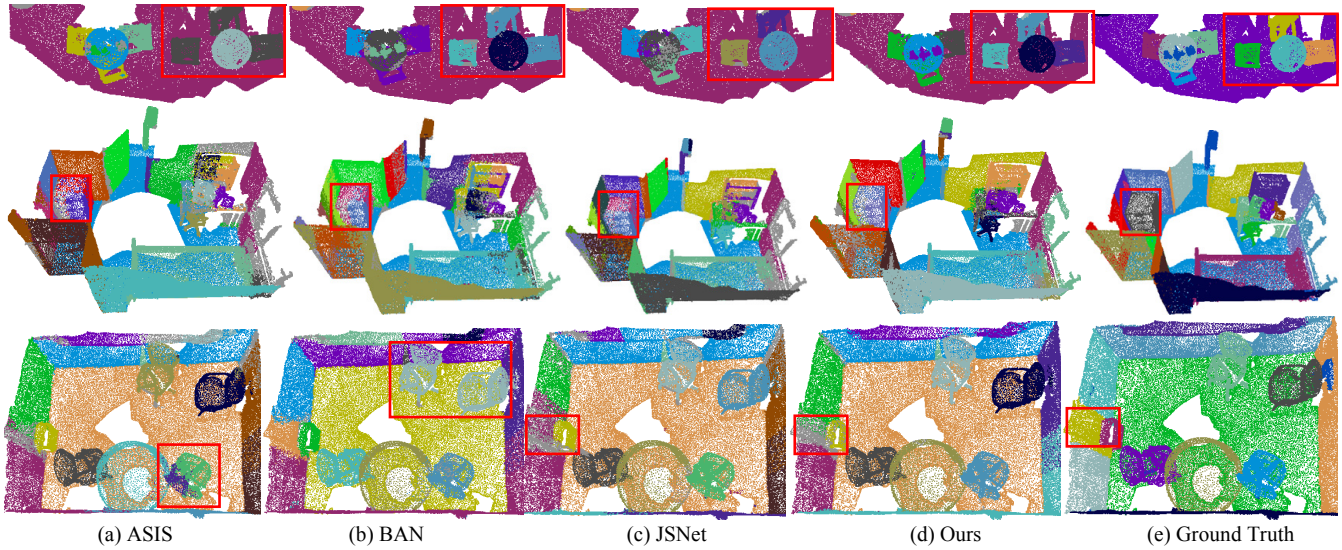
As shown in Table VII, the ABJS, PFE and AP modules could revise the fundamental results of semantic segmentation and instance segmentation. Equipped with different components upon the baseline network, the experimental results show that our proposed ABJNet outperforms the baseline to a large extent.

We can find that with our ABJS module, there are 8.2% gains on mPrec and 1.3% gains on mIoU compared with the baseline. The performance of instance segmentation and semantic segmentation is improved with the ABJS module, which suggests merging instance features for semantic segmentation in our way is very efficient. Semantic awareness can help the instance predictions and improved instance predictions could assign more accurate semantic labels in semantic-instance segmentation tasks. With the PFE module added to the network with the ABJS module, the improvement is more significant for two metrics: 8.8% mPrec and 4.5% mIoU. Finally, compared with the baseline, our method has a large improvement in all evaluation metrics, achieving 57.8% mPrec and 59.0% mIoU for instance and semantic segmentation tasks, respectively.



(a) ASIS (b) BAN (c) JSNet (d) Ours (e) Ground Truth

Figure 6. Comparison results of ASIS, BAN, JSNet and our method on semantic segmentation task on ScanNet V2



(a) ASIS (b) BAN (c) JSNet (d) Ours (e) Ground Truth

Figure 7. Comparison results of ASIS, BAN, JSNet and our method on instance segmentation task on ScanNet V2

TABLE VI. COMPARISON OF COMPUTATION TIME, GPU MEMORY AND PERFORMANCE BASED ON AREA5 OF S3DIS DATASET

	Methods	Train		Test		mPrec (%)	mIoU (%)
		Time(m)	Memory(MB)	Time(m)	Memory(MB)		
Area 5	ASIS[3]	12.7	7847	32.6	4944	55.3	53.4
	BAN[23]	13.2	7029	32.8	5024	56.7	55.2
	Ours	9.8	7262	29.7	5015	57.8	59.0

TABLE VII. ABLATION STUDY ON THE S3DIS DATASET IN AREA 5

Component			Instance segmentation				Semantic segmentation		
ABJS	PFE	AP	mCov (%)	mWCov (%)	mPrec (%)	mRec (%)	mAcc (%)	mIoU (%)	oAcc (%)
×	×	×	45.2	48.2	49.3	41.6	62.1	54.3	87.1
√	×	×	48.9	52.0	57.5	45.4	63.2	55.6	87.7
√	√	×	50.0	53.1	58.1	46.7	65.6	58.8	89.3
√	√	√	50.3	53.5	57.8	48.2	65.7	59.0	89.4

VI. CONCLUSION

The 3D semantic and instance segmentation aim to detect specific informative region represented by sets of smallest units in the scene. Both of them have wide applications in scene understanding such as autonomous driving and intelligent robot. For autonomous driving, scene segmentation is a prerequisite for the vehicle to effectively obtain the drivable area and obstacles on the road. It plays a leading role in decision-making, trajectory planning, and control in difficult environments with other traffic participants and obstacles. For the intelligent robot, the result of semantic and instance segmentation helps the robot to establish the 3D semantic map and perceive the semantic categories of objects in the real environment. It lays a good foundation for path planning and high-level decision-making tasks.

In this paper, we propose an attention-based network named ABJNet for semantic and instance segmentation of point clouds. The proposed network could be used to learn instance-aware semantic feature maps and semantic-aware instance feature embedding which are more discriminative and accurate for 3D point cloud segmentation. An attention-based feature fused module is designed to collaborate and mutually reinforce instance and semantic segmentation.

Experiments on S3DIS and ScanNet V2 datasets demonstrate the effectiveness and efficiency of the proposed ABJNet. However, the segmentation of indoor scenes is still a challenging task due to its high occlusion, high clutter, and large variability. In addition, the segmentation methods based on convolutional neural network imply the time-consuming collection of training data. When the dataset is small, it is difficult to collect sufficient models for training. The training process is also time-consuming. In the future work, a lightweight convolutional neural network with high computational efficiency and small parameter should be considered. It can be deployed in embedded devices and has a broad application prospect in the point cloud real-time processing.

REFERENCES

[1] J. Wu, J. Jiao, Q. Yang, Z. Zha, X. Chen, "Ground-aware point cloud semantic segmentation for autonomous driving," Proceedings of the 27th ACM International Conference on Multimedia. 2019, pp.971-979. doi:10.1145/3343031.3351076

[2] Y. Nie, J. Hou, X. Han and M. Nießner, "RfD-Net: Point scene understanding by semantic instance reconstruction," IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4606-4616. doi:10.1109/CVPR46437.2021.00458

[3] X. Wang, S. Liu, X. Shen, C. Shen, J. Jia, "Associatively segmenting instances and semantics in point clouds," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp.4096-4105. doi:10.1109/CVPR.2019.00422

[4] C. R. Qi, L. Yi, H. Su, L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," International Conference on Neural Information Processing Systems, 2017, pp. 5105–5114. doi:10.5555/3295222.3295263

[5] S. Qiu, S. Anwar, N. Barnes, "Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp.1757-1767. doi:10.1109/CVPR46437.2021.00180

[6] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, F.Y. Wang, "SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 14499-14508. doi:10.1109/CVPR46437.2021.01427

[7] Y. Su, W. Liu, Z. Yuan, et al., "DLA-Net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds," Pattern Recognit. 123: 108372, 2022. doi:10.1016/j.patcog.2021.108372

[8] J. Hou, A. Dai, M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, pp. 4421-4430, doi:10.1109/CVPR.2019.00455

[9] L. Yi, W. Zhao, H. Wang, M. Sung, L. Guibas, "GSPN: Generative shape proposal network for 3D instance segmentation in point cloud," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 3942-3951. doi:10.1109/CVPR.2019.00407

[10] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, "Learning object bounding boxes for 3D instance segmentation on point clouds," Advances in neural information processing systems, 2019, 32. doi:10.48550/arXiv.1906.01140

[11] F. Zhang, C. Guan, J. Fang, S. Bai, R. Yang, P. Torr, V. Prisacariu, "Instance segmentation of Lidar point clouds," IEEE International Conference on Robotics and Automation. 2020, pp.9448-9455. doi:10.1109/ICRA40945.2020. 9196622

[12] W. Wang, R. Yu, Q. Huang, U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition. 2020, pp.2569-2578. doi:10.1109/CVPR.2018.00272

[13] C. R. Qi, H. Su, K. Mo, L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp.77-85. doi:10.1109/CVPR.2017.16

[14] C. Liu, Y. Furukawa, "Masc: Multi-scale affinity with sparse convolution for 3D instance segmentation," arXiv preprint arXiv:1902.04478, 2019. doi:arxiv.org/abs/1902.04478

[15] B. Graham, M. Engelcke, L. Van Der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp.9224-9232. doi:10.1109/CVPR.2018.00961

[16] Z. Liang, M. Yang, H. Li, C. Wang, "3D instance embedding learning with a structure-aware loss function for point cloud segmentation," IEEE Robotics and Automation Letters. vol.5, no.3, pp.4915-4922, 2020. doi:10.1109/LRA.2020.3004802

[17] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.5, pp.603-619, 2002. doi:10.1109/34.1000236

[18] L. Jiang, H. Zhao, S. Shi, S. Liu, C. W. Fu, J. Jia, "Pointgroup: Dual-set point grouping for 3D instance segmentation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 4866-4875. doi:10.1109/CVPR42600.2020.00492

[19] T. He, C. Shen, A. Hengel. "Dynamic Convolution for 3D point cloud instance segmentation," arXiv preprint arXiv:2107.08392, 2021. doi:10.48550/arXiv.2107.08392

[20] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, "Deep learning for 3D point clouds: A survey," IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 12, pp. 4338-4364, 2020. doi:10.1109/TPAMI.2020.3005434

[21] Q. H. Pham, T. Nguyen, B. S. Hua, G. Roig, S. K. Yeung, "JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp.8827-8836. doi:10.1109/CVPR.2019.00903

[22] L. Zhao, W. Tao, "JSNet: Joint instance and semantic segmentation of 3D point clouds," Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, no. 7, pp. 12951-12958, 2020. doi:10.1609/aaai.v34i07.6994

[23] G. Wu, Z. Pan, P. Jiang, C. Tu, "Bi-Directional attention for joint instance and semantic segmentation in point clouds," Proceedings of the Asian Conference on Computer Vision, 2020, pp. 1-17. doi:10.48550/arXiv.2003.05420

[24] F. Chen, F. Wu, G. Gao, Y. Ji, J. Xu, G. Jiang, X. Jing, "JSPNet: Learning joint semantic & instance segmentation of point clouds via feature self-similarity and cross-task probability," Pattern Recognit. vol. 122, no. 108250, 2022. doi:10.1016/j.patcog.2021.108250

[25] C. Chen, L. Z. Fragonara, A. Tsourdos, "GAPNet: Graph attention based point neural network for exploiting local feature of point cloud," arXiv preprint arXiv:1905.08705, 2019. doi:10.48550/arXiv.1905.08705

[26] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, "RandLA-Net: efficient semantic segmentation of large-scale point clouds," Proceedings of the Computer Vision and Pattern Recognition, 2020, pp. 11105-11114. doi:10.1109/CVPR42600.2020.01112

[27] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, "3D semantic parsing of large-scale indoor spaces," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 1534-1543. doi:10.1109/CVPR.2016.170

[28] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 2432-2443. doi:10.1109/CVPR.2017.261

[29] L. Du, J. Tan, X. Xue, L. Chen, "3DCFS: Fast and robust joint 3D semantic-instance segmentation via coupled feature selection," IEEE International Conference on Robotics and Automation. 2020, pp. 6868-6875. doi:10.1109/ICRA40945.2020.9197242

[30] M. Zhong, G. Zeng, "Joint Semantic-Instance Segmentation of 3D point clouds: Instance separation and semantic fusion," 25th International Conference on Pattern Recognition. 2021, pp. 6616-6623. doi:10.1109/ICPR48806.2021.9412532