Bi-Directional Attention for Joint Instance and Semantic Segmentation in Point Clouds

Guangnan Wu[1][0000−0002−0841−5791], Zhiyi Pan[1][0000−0002−0635−0349], Peng Jiang[1⋆][0000−0002−7342−7940], and Changhe Tu[1⋆][0000−0002−1231−3392]

[1]Shandong University, China.
{wuguangnan1006, panzhiyi1996, sdujump, changhe.tu}@gmail.com

Abstract. Instance segmentation in point clouds is one of the most fine-grained ways to understand the 3D scene. Due to its close relationship to semantic segmentation, many works approach these two tasks simultaneously and leverage the benefits of multi-task learning. However, most of them only considered simple strategies such as element-wise feature fusion, which may not lead to mutual promotion. In this work, we build a Bi-Directional Attention module on backbone neural networks for 3D point cloud perception, which uses similarity matrix measured from features for one task to help aggregate non-local information for the other task, avoiding the potential feature exclusion and task conflict. From comprehensive experiments on the three prevalent datasets, as well as ablation and efficiency studies, the superiority of our method is verified. Moreover, the mechanism of how bi-directional attention module helps joint instance and semantic segmentation is also analyzed.

1 Introduction

Among the tasks of computer vision, instance segmentation is one of the most challenge ones which requires understanding and perceiving the scene in unit and instance level. Notably, the vast demands for machines to interact with real scenarios, such as robotics and autonomous driving [1, 2], make the instance segmentation in the 3D scene to be the hot research topic.

Though much progress has been made, 3D instance segmentation still lags far behind its 2D counterpart [3–8]. Unlike the 2D image, the 3D scene can be represented by many forms, such as multi-view projection images [9–13], volumes [14–17], and point clouds.

Since point clouds could represent a 3D scene more compactly and intuitively, and thus became more popular and drew more attention recently. The proposed PointNet [18] and some following works [19–28] could process the raw point clouds directly, achieving remarkable performance on 3D classification and part segmentation tasks. The success brings the prospect for more fine-grained perception tasks in 3D point clouds, such as instance segmentation.

Instance segmentation in point clouds requires distinguishing category and instance belonging to each point. The most direct way is to regress further each

⋆ Corresponding author.

instance's bounding box based on the semantic segmentation results, such as [29–31].

Due to the close relationship between instance segmentation and semantic segmentation, most of the recent works approach these two tasks simultaneously and use deep neural networks with two sub-branches for the two tasks, respectively [32–34]. Among them, many take feature fusion strategy letting features for one task promote the other task. However, in fact, the features of the two tasks are not completely compatible with each other. While points belong to different semantics must belong to different instances, points in the different instances are not necessarily of the different semantics. Obviously, directly concatenating or adding these two kinds of features in the model may lead to task conflict.

Actually, with simple element-wise feature fusion way such as concatenating and adding, only semantic features could always help distinguish instances in all the cases.

This situation poses a question, do we still need instance features for semantic segmentation and how to make these two tasks mutually promoted? In this work, we invest another way to incorporate features for semantic and instance segmentation. Instead of explicitly fusing features, we use similarity information implied in features for one task to assist the other task. Specifically, we first measure pair-wise similarity on semantic features to form the semantic similarity matrix, with which we propagate instance features. The propagation operation computes the response at a point as a weighted sum of the features at all points with semantic similarity as weight. Finally, the responses are further concatenated to the original instance features for instance segmentation. The same steps are also conducted in another direction that computing instance similarity matrix to propagate semantic features for semantic segmentation. The propagation operation could aggregate non-local information and is also referred to as attention [35–38]. Therefore, we name this kind of module as Bi-Directional Attention and call our networks as BAN.

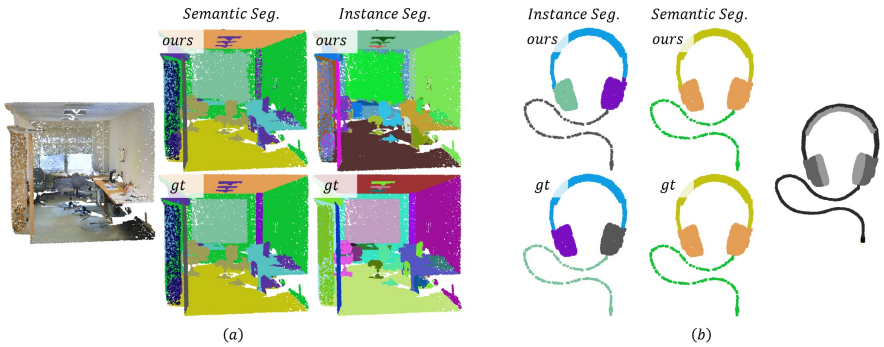

Fig. 1. Instance and semantic segmentation in point clouds using BAN. (a) Results on the S3DIS dataset, (b) Results on the PartNet dataset.

The help of Bi-Directional Attention module lies in the following aspects. First, for aggregation operation applied to instance features for instance segmentation, semantic similarity matrix would help push instance features belonging to the different semantic apart. Though it will also pull instance features belonging to the same semantic together, the concatenated original instance features could still guarantee the difference distinguishable. Second, for aggregation operation applied on semantic features for semantic segmentation, instance similarity matrix would let semantic within each instance more consistent, thus improve the detail delineation. In addition to the positive effects when using bi-directional attention in a forward manner, the operation will also be good for back-propagating uniform gradients within the same semantic or instance. Consequently, our Bi-Directional Attention module could aggregate the features more properly and avoid potential task conflict.

We compare our BAN to state-of-the-art methods on prevalent 3D point cloud datasets, including S3DIS [39], PartNet [40] and ScanNetV2 [41]. Some instance and semantic segmentation results is shown in Fig. 1. In experiments, our method demonstrates consistent superiority according to most of the evaluation metrics. Moreover, we conduct detailed ablation, mechanism and efficiency studies, which suggest that the similarity matrices truly reflect the required pair-wise semantic and instance similarities without too much computation complexity increase.

With attention operations from two directions together sequentially, BAN we can reach the best performance. Our code has been open sourced.

2 Related Works

Here, we mainly focus on methods that are most relevant to ours.

As well known, PointNet [18], for the first time, used neural networks to perceive point clouds and showed leading results on classification and semantic segmentation. However, it has difficulties in capturing local and fine-grained features. Correspondingly, many sequential works proposed to address this problem, such as [19–27].

Recently, instance segmentation in point clouds has drawn intense attention. Many works have been proposed and could be divided into two types in general, proposal-based and proposal-free. The former ones usually follow the scheme of Mask R-CNN [4] in 2D images, leading to a two-stage training, such as 3D-SIS [29] and GSPN [30]. Unlike them, BoNet [31] follows the one-stage scheme and regresses the bounding box directly. Nevertheless, the bounding box sometimes contains multiple objects or just a part of an object, making proposal-based methods hard to delineate the instance precisely. In contrast, the latter ones, *e.g.*, SGPN [42], 3D-BEVIS [43], JSIS3D [33], ASIS [32] and JSNet [34], directly produce representations to estimate the semantic categories and cluster the instance groups for each element, correspondingly, obtain more fine-grained perception.

It is worth to note that, whether for semantic segmentation or instance segmentation in 2D images, capturing long-range dependency and non-local information had been the consensus approach to improve accuracy. For this purpose, attention has been invented in [35], and become basic operation that applied prevalently [38, 37]. However, this operation has not been well studied for 3D point cloud perception.

3 Motivation

In this work, we intend to propose a proposal-free type of joint instance and semantic segmentation method in point clouds. For this task, the key issue is how to incorporate the features of semantic and instance efficiently for mutual benefits. In view of the close relationship between instance and semantic segmentation, JSNet [34] fuses semantic and instance features to each other by simple aggregation strategies such as element-wise add and concatenate operations. In this way, the problem can be formalized as the following equations:

$$\begin{aligned} \mathcal{F}(\alpha(S_a, I_a)) \rightarrow C_a, \qquad \mathcal{F}(\alpha(S_b, I_b)) \rightarrow C_b, \\ \mathcal{H}(\alpha(S_a, I_a)) \rightarrow G_a, \qquad \mathcal{H}(\alpha(S_b, I_b)) \rightarrow G_b, \end{aligned} \tag{1}$$

where S_i and I_i represent semantic and instance features of point i respectively, and C_i and G_i are the semantic category and instance group of point i. α is some simple feature aggregating method. We use $\mathcal{F}$ and $\mathcal{H}$ to represent mapping functions for semantic and instance segmentation, respectively.

Ideally, there are three cases for two points a and b: (1) C_a=C_b and G_a≠G_b; (2) C_a=C_b and G_a=G_b; (3) C_a≠C_b and G_a≠G_b. In the first case, for semantic segmentation $\mathcal{F}$, aggregating S and I by α will make responses $\alpha(S_a, I_a)$ and $\alpha(S_b, I_b)$ far away. Thus C_a and C_b are hard to keep consistent, which is contrary to the case setting. In the second case, both $\mathcal{F}$ and $\mathcal{H}$ could get promoted by aggregating features of the same instance by α. The third case will not be considered when aggregating feature, because a and b are not relevant in either semantic or instance. So, with the simple aggregation strategy adopted by JSNet [34], there is a potential risk of task conflict in some specific cases.

Some works get rid of task conflict problem by introducing more complex feature aggregation strategies. JSIS3D [33] uses multi-value conditional random field to fuse semantic and instance, but it requires some approximation to optimize. ASIS [32] uses KNN to assemble more instance features from the neighborhood to each point and make the assembled feature more robust, but the KNN operation is non-differentiable and will break the back-propagation chain. The use of KNN in this work could be considered as proto non-local operation.

In summary, simple feature aggregation strategies such as element-wise add and concatenation will bring task conflict potential while other more complex feature aggregation strategies are far more satisfying.

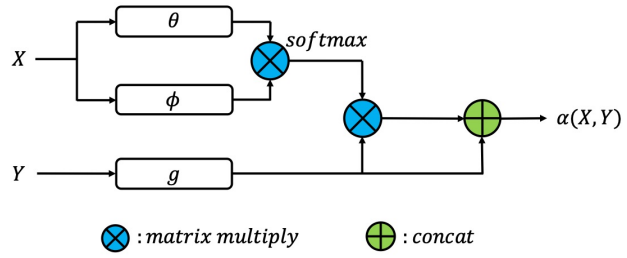

Fig. 2. Attention operation.

4 Methodology and Implementation

4.1 Methodology

As discussed in Sec. 3, for semantic segmentation, just adding or concatenating instance feature to semantic feature will be problematic. It poses a question, does instance feature has any help for semantic segmentation?

Here we suggest a way to use similarity information implied in the instance features to help semantic segmentation without any harm. To be specific, we propose to adjust the point's semantic feature as the weighted sum of semantic features of points belong to the same instance (with similar instance features). This way would make the semantic features robust and consistent within each instance, which will promote the details delineation.

To enable this function and take advantage of similar information in the instance features, we design the aggregation operation as:

$$\begin{aligned} \alpha(X, Y) &= \{P \cdot g(Y), Y\}, \\ P &= softmax(\theta(X)\phi(X)^T), \end{aligned} \tag{2}$$

where X and Y represent two kinds of features of size $N \times N_X$ and $N \times N_Y$ respectively (N is point number and N_i is number of channels for feature i). θ, ϕ and g are functions to re-weighted sum values in feature dimension with learned weights. Here, α is the concat operation. We measure similarities by inner-product of $\theta(X)$ and $\phi(X)$, which results into a matrix of size $N \times N$. We further apply $softmax$ on each row to get transition matrix which is our final similarity matrix P.

When X is instance features, and Y is semantic features, this operation propagates semantic features to other points by instance similarity matrix and the adjusted semantic features $P \cdot g(Y)$ will be more uniform in each instance than the original Y. Since there is no explicit element-wise adding or concatenating between semantic and instance features, using the final aggregation result $\alpha(X, Y)$ for semantic segmentation will not have the problems mentioned in the last section. Besides, this aggregation operation has the non-local characteristic naturally. For these reasons, we will also use it to fuse semantic features for instance segmentation. In other words, we will conduct another aggregation

operation with X as semantic features and Y as instance features for instance segmentation. It is worth to note that though in this case, Eq. 2 will tend to pull instance features belonging to the same semantic together, the concatenated original instance features could still guarantee the difference distinguishable.

The above-defined operation has a similar form as attention in [35]. However, we have two of them with different architecture and goals. We have two different data flow directions, to aggregate semantic and instance features with the help of similarity inherent in features. Consequently, we name the proposed module as the Bi-Directional Attention module. The architecture of our module is illustrated in Fig. 2.

4.2 Implementation

Networks By connecting the Bi-Directional Attention module to the end of the feature extracting backbone, we have the Bi-Directional Attention networks (BAN), which uses two attention operations to achieve information transmission and aggregation between instance branch and semantic branch. The full pipeline of our networks is illustrated in Fig. 3.

Our BAN is composed of a shared encoder, and two parallel decoders to produce representations for estimating the semantic categories and clustering the instance groups. Specifically, our backbone is PointNet++ [19]. Given input point clouds of size N, the backbone first extracts and encodes them into feature matrix which further decoded to semantic feature matrix S of size $N \times N_S$ and instance feature matrix I of $N \times N_I$.

The Bi-Directional Attention module takes these two feature matrices as input and will conduct two attention operations as defined by Eq. 2. We name the attention operation that computes semantic similarity matrix applied to instance features for instance segmentation as STOI, and attention operation that computes instance similarity matrix applied to semantic features for semantic segmentation as ITOS. The output of STOI is further passed to some simple fully connected layers (FC) to produces instance embedding space (of size $N \times N_E$), while the output of ITOS is further passed to some simple fully connected layers (FC) to give semantic prediction (of size $N \times N_C$). To get the instance groups, we cluster the produced instance embedding space by mean-shift method [44].

There are three kinds of sequences to conduct STOI and ITOS, and they are STOI first, ITOS first, and simultaneously. Here we use STOI first because we will use pixel-level regression loss for semantic segmentation and discriminative loss for instance segmentation, and we believe semantic features will converge faster than instance features. So, semantic features will give instance segmentation task more help at the beginning. This assumption will be verified in our ablation study in Sec. 5.

Loss Function Our loss function $\mathcal{L}$ has two parts, semantic segmentation loss $\mathcal{L}_{sem}$ and instance segmentation loss $\mathcal{L}_{ins}$. These two parts are optimized at the same time:

$$\mathcal{L} = \mathcal{L}_{sem} + \mathcal{L}_{ins}. \tag{3}$$

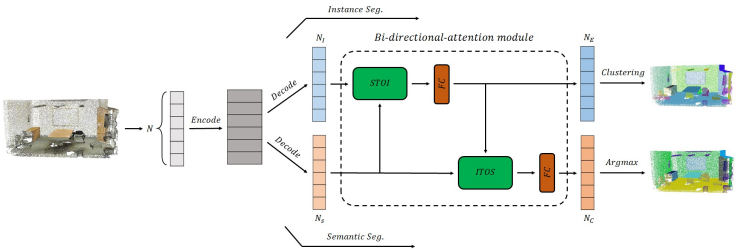

Fig. 3. The pipeline of proposed Bi-Directional Attention Networks (BAN).

We use cross-entropy loss for $\mathcal{L}_{sem}$, and choose discriminative loss for 2D images in [8] as $\mathcal{L}_{ins}$. The discriminative loss has been extended to 3D point clouds and used by many works [32–34]. $\mathcal{L}_{ins}$ will penalize the grouping of the points across different instances and bring the points belonging to the same instance closer in the embedding space. For the details, please check the supplementary.

Derivative Analysis The above sections have explained how our module gives help in a forward manner. Here we further analyze the back-propagation of proposed Eq. 2. To simplify the problem, we first give a simple version of Eq. 2 without softmax, re-weight functions, and concatenation of original features:

$$Z = XX^TY. \tag{4}$$

where Z is the output of simplified attention operation.

In this case, the derivatives with respect to feature X and Y are:

$$\begin{aligned} vec(d\mathcal{L}) &= (\frac{\partial \mathcal{L}}{\partial Z})^T vec(dZ) \\ &= (\frac{\partial \mathcal{L}}{\partial Z})^T [vec(dXX^TY) + vec(XdX^TY)] \\ &= (\frac{\partial \mathcal{L}}{\partial Z})^T [(X^TY)^T \otimes E_N + (Y^T \otimes X)K_{NN_X}] vec(dX) \\ \frac{\partial \mathcal{L}}{\partial X} &= [(X^TY) \otimes E_N + K_{NN_X}(Y \otimes X^T)] \frac{\partial \mathcal{L}}{\partial Z} \end{aligned} \tag{5}$$

$$\begin{aligned} vec(d\mathcal{L}) &= (\frac{\partial \mathcal{L}}{\partial Z})^T vec(dZ) \\ &= (\frac{\partial \mathcal{L}}{\partial Z})^T vec(XX^TdY) \\ &= (\frac{\partial \mathcal{L}}{\partial Z})^T (E_{N_Y} \otimes XX^T) vec(dY) \\ \frac{\partial \mathcal{L}}{\partial Y} &= (E_{N_Y} \otimes XX^T) \frac{\partial \mathcal{L}}{\partial Z} \end{aligned} \tag{6}$$

where $vec()$ means matrix vectorization and $\otimes$ represents Kronecker Product, E is identity matrix and K is commutation matrix.

It can be seen, the similarity matrices also appear in $\frac{\partial \mathcal{L}}{\partial X}$ and $\frac{\partial \mathcal{L}}{\partial Y}$. As for XX^T in $\frac{\partial \mathcal{L}}{\partial Y}$, it will make the gradients uniform and robust within a similar region defined by X (semantic or instance), thus help optimization. As for X^TY, it computes similarities between different features of X and Y other than points and provides another crucial information to extract robust and useful gradients.

In summary, the proposed Bi-Directional Attention module not only help joint instance and semantic segmentation by transmitting and aggregating information between instance features and semantic features, and also be good for back-propagating uniform and robust gradients.

5 Experiments

5.1 Experiments Setting

Datasets We train and evaluate methods on three prevalent used datasets. *Stanford 3D Indoor Semantics (S3DIS) [39]* contains 3D scans in 6 areas including 271 rooms. Each scanned 3D point is associated with an instance label and a semantic label from 13 categories. *PartNet [40]* contains 573,585 fine-grained part instances with annotations and has 24 object categories. *ScanNetV2* [41] is an RGB-D video dataset containing 2.5 million views in more than 1500 scans.

Evaluation Metrics For semantic segmentation, we compare our BAN with others by overall accuracy (oAcc), mean accuracy (mAcc), and mean IoU (mIoU).

As for instance segmentation, coverage (Cov) and weighted coverage (WCov) [45–47] are adopted.

Cov and Wcov are defined as:

$$Cov(\mathcal{G}, \mathcal{O}) = \sum_{i=1}^{|\mathcal{G}|} \frac{1}{|\mathcal{G}|} \max_j IoU(r_i^G, r_j^O) \tag{7}$$

$$WCov(\mathcal{G}, \mathcal{O}) = \sum_{i=1}^{|\mathcal{G}|} \frac{1}{|\mathcal{G}|} \omega_i \max_j IoU(r_i^G, r_j^O), \quad \omega_i = \frac{|r_i^G|}{\sum_k |r_k^G|} \tag{8}$$

where ground-truth is denoted as $\mathcal{G}$ and prediction is denoted as $\mathcal{O}$, $|r_i^G|$ is the number of points in ground-truth i. Besides, the classical metrics mean precision (mPrec), and mean recall (mRec) with IoU threshold 0.5 are also reported.

Hyper-parameters To optimize the proposed BAN, we use Adam optimizer [48] with batch size 12 and set initial learning rate as 0.001 following the "divided by 2 every $300k$ iterations" learning rate policy. During training, we use the default parameter setting in [8] for $\mathcal{L}_{ins}$. At test time, bandwidth is set to 0.6 for mean-shift clustering. BlockMerging algorithm proposed by SGPN [42] is used to merge instances from different blocks. Please check supplementary for the training and testing details on the three datasets.

Table 1. Instance segmentation results on S3DIS dataset.

Method	Backbone	mCov	mWCov	mPrec	mRec
Test on 6-fold cross-validation					
PointNet	PointNet	43.0	46.3	50.6	39.2
PointNet++	PointNet++	49.6	53.4	62.7	45.8
SGPN	PointNet	37.9	40.8	38.2	31.2
ASIS	PointNet++	51.2	55.1	63.6	47.5
BoNet	PointNet++	46.0	50.2	**65.6**	47.6
JSNet	PointNet++	46.4	50.3	58.9	43.0
Ours	PointNet++	**52.1**	**56.2**	63.4	**51.0**

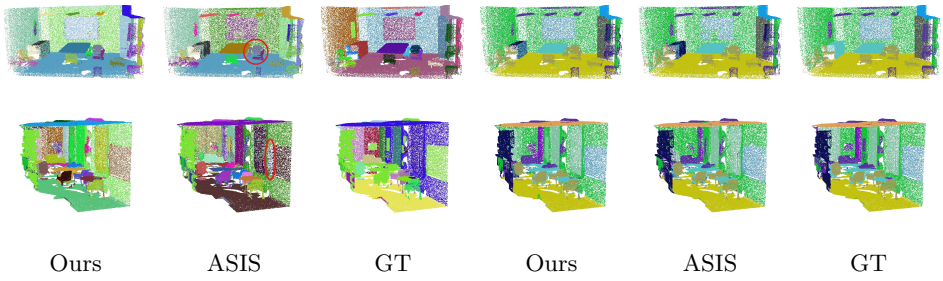

Fig. 4. Visual comparison of instance and semantic segmentation results on the S3DIS dataset. The first three columns are the instance segmentation results, while the last three columns show semantic segmentation results.

5.2 S3DIS Results

In this section, we will compare our method (BAN) with other state-of-the-art methods, and the reported metric values are either from their papers or implemented and evaluated by ourselves when not available.

Instance segmentation In Tab. 1, six methods are compared, including PointNet[18], PointNet++[19], SGPN[42], ASIS[32], BoNet[31] and our BAN. It's worth to note that, PointNet++ has the same architecture and settings as ours except the Bi-Directional Attention module, and thus can be treated as baseline. PointNet is similar to PointNet++ except the backbone. It can be seen, our BAN outperforms baseline (PointNet++) on all the metrics, and demonstrates significant superiority compared with others.

Semantic segmentation Since SGPN[42] and BoNet[31] do not provide semantic segmentation results. For semantic segmentation, we only compare PointNet[18], PointNet++[19] and ASIS[32]. The evaluation results are shown in Tab. 2, from mAcc, mIoU, and oAcc, our method achieves the best performance consistently.

Table 2. Semantic segmentation results on S3DIS dataset.

Method	Backbone	mAcc	mIoU	oAcc
Test on 6-fold cross-validation				
PointNet	PointNet	60.7	49.5	80.4
PointNet++	PointNet++	69.0	58.2	85.9
ASIS	PointNet++	70.1	59.3	86.2
JSNet	PointNet++	65.5	56.3	85.5
Ours	PointNet++	**71.7**	**60.8**	**87.0**

Visual Comparison We show some visual results of semantic and instance segmentation methods in Fig. 4. From results, we can see ours are more accurate and uniform compared with ASIS [32], especially for instance segmentation as marked by red circles. We believe it is because of the applying of attention operations and the introduction of non-local information. The more studies of attention mechanisms are in Sec. 6.

Table 3. Instance segmentation results on PartNet dataset.

Method	Backbone	mCov	mWCov	mPrec	mRec
PointNet++	PointNet++	42.0	43.1	51.2	44.7
ASIS	PointNet++	39.3	40.2	49.9	42.8
Ours	PointNet++	**42.7**	**44.2**	**52.8**	**45.3**

5.3 PartNet Results

In addition to object instance segmentation in indoor scenes, we further evaluate our method on part instance segmentation in objects using the PartNet dataset. This task is more fine-grained and thus requires more perception ability to understand the similarity between points.

The semantic and instance segmentation scores are listed in Tab. 3, 4. We can see that the performance has a significant drop compared with the S3DIS. This is because the dataset contains many kinds of small semantic parts, which are difficult to perceive and predict, causing low semantic mIoU and instance mCov but relative high semantic oAcc.

For this kind of dataset with small semantic parts, ASIS [32] with KNN is difficult to adapt by a fixed range control parameter. However, with the Bi-Directional Attention module, our method could compute the similarities between any of two points and achieves better results.

The visual results on PartNet are shown in Fig. 5. Our method demonstrates obvious advantages compared with ASIS [32], and produces more accurate instance and semantic segmentation, especially for some small parts as marked by red circles. For other methods compared in S3DIS, their performance is not evaluated in this section, because we do not have their code or statistic report.

Table 4. Semantic segmentation results on PartNet dataset.

Method	Backbone	mAcc	mIoU	oAcc
PointNet++	PointNet++	53.4	43.4	78.4
ASIS	PointNet++	50.6	40.2	76.7
Ours	PointNet++	**56.1**	**44.9**	**80.3**

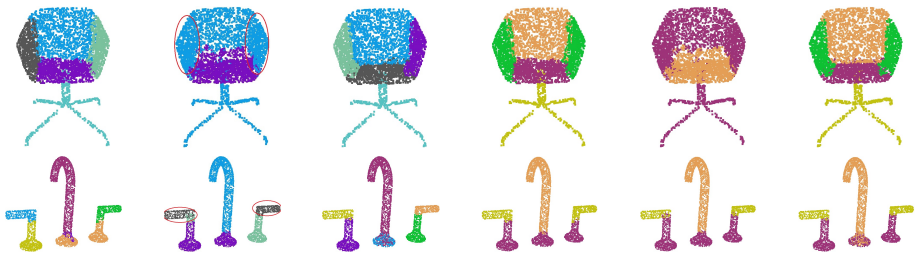

Fig. 5. Visual comparison of instance and semantic segmentation results on the PartNet dataset. Columns are arranged as Fig. 4.

5.4 ScanNetV2 Results

Finally, we evaluate the performance on the ScanNetV2 which is the biggest indoor 3D point cloud dataset by now. The quantitative results are listed in Tab. 5 and Tab. 6, while the qualitative results are shown in Fig. 6. We only evaluate the methods we have code or corresponding statistic report. All the results have verified the superiority of our method in the large scale dataset.

Table 5. Instance segmentation results on ScanNetV2 dataset.

Method	Backbone	mCov	mWCov	mPrec	mRec
PointNet++	PointNet++	39.0	40.1	46.0	40.1
ASIS	PointNet++	39.1	40.4	46.3	40.5
Ours	PointNet++	**40.4**	**41.7**	**48.2**	**42.2**

6 Discussion

In this section, we intend to show more evidence to justify the design and the mechanism of the proposed Bi-Directional Attention module.

6.1 Ablation study

As mentioned in Sec. 4.2, there are three kinds of sequences to conduct STOI and ITOS in our Bi-Directional Attention module, and we gave an assumption to

Table 6. Semantic segmentation results on ScanNetV2 dataset.

Method	Backbone	mAcc	mIoU	oAcc
PointNet++	PointNet++	58.3	47.1	82.3
ASIS	PointNet++	58.5	46.5	81.9
Ours	PointNet++	**60.8**	**48.8**	**82.7**

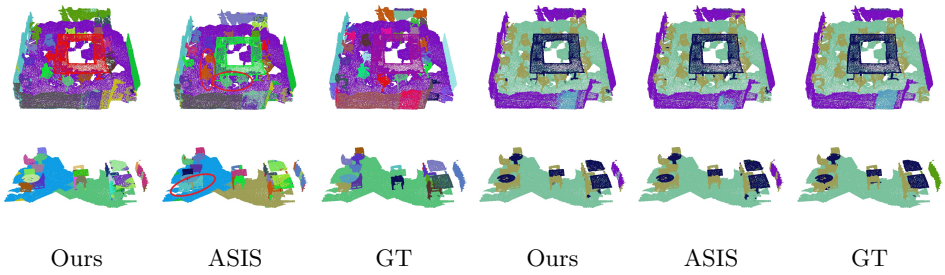

Fig. 6. Visual comparison of instance and semantic segmentation results on the ScanNetV2 dataset. Columns are arranged as Fig. 4.

decide our design. Here, we will verify our choice and further prove the necessity to have both STOI and ITOS.

In Tab. 7, we give five rows of results for instance and semantic segmentation with different combinations and order of STOI and ITOS. The experiments are conducted on Area 5 of S3DIS [39]. We can see, by introducing STOI, the instance segmentation gets boosted. With ITOS, both instance and semantic segmentation demonstrate certain improvement, which suggests fusing instance features for semantic segmentation in our way is very effective. Moreover, considering the potential task conflict when using simple element-wise feature aggregation strategies such as adding and concatenating, the improvement is more significant. Finally, with both STOI and ITOS, and STOI first, we achieve the best results. But, with an inverse order that ITOS first, the performance shows a large drop, even worse than results without STOI and ITOS. This phenomenon verified the importance of order to conduct STOI and ITOS and is worth to be studied further in the future.

Further, we test performance when $X = Y$ in Eq. 2 where our Bi-Directional Attention module is degraded to two independent self-attention operations [35]. The result is listed in the last row of Tab. 7. Obviously, without feature fusing, self-attention is not comparable to our method.

6.2 Mechanism Study

Here, we visualize the learned instance and semantic similarity matrices P defined in Eq. 2 to study and verify their mechanism. The similarity matrix is the key functional unit, which builds the pair-wise similarities and uses to weighted-sum non-local information. A good instance similarity matrix should accurately

Table 7. Results of all ablation experiments on Area 5 of S3DIS.

Ablation		Instance segmentation				Semantic segmentation		
STOI	ITOS	mCov	mWCov	mPrec	mRec	mAcc	mIoU	oAcc
×	×	46.0	49.1	54.2	43.3	62.1	53.9	87.3
✓	×	47.1	50.1	55.3	43.6	61.2	53.4	87.0
×	✓	47.4	50.3	54.0	43.4	62.0	54.7	87.8
✓	✓	**49.0**	**52.1**	**56.7**	**45.9**	**62.5**	**55.2**	87.7
Inverse order		46.3	49.4	53.5	41.5	**62.5**	55.1	**87.9**
Self-attention		45.4	48.6	53.3	43.6	**62.5**	55.1	**87.9**

reflect the similarity relationship between all of the points, so P are of size $N \times N$. When the instance/semantic similarity matrix trained well, it will help generate uniform and robust semantic/instance features. Besides, good instance and semantic similarity matrices will also benefit the back-propagation process, as stated in Sec. 4.2.

In Fig. 7, for trained networks and each sample, we select the same row from instance similarity matrix and semantic similarity matrix, respectively, then reshape the row vector to the 3D point cloud. So, the value of each point here represents the similarity to the point corresponding to the selected row. For better visualization, we binarize the 3D point cloud to divide points into two groups, similar points (green) and dissimilar points (blue) and marked the point corresponding to the selected row by red circle. Each sample of Fig. 7 has two chairs in the scenes. We can see that the semantic similarity matrix could basically correctly reflect the semantic similarities, and the instance similarity matrix could highlight most of the points in the same instance.

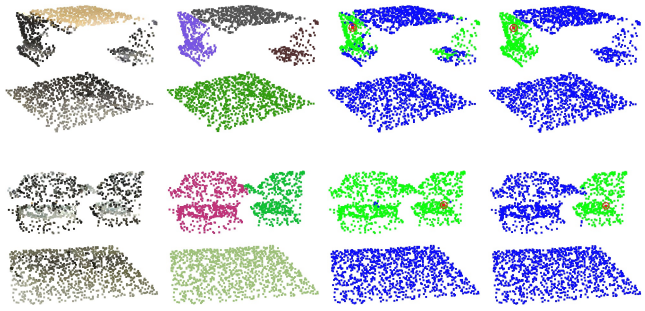

Fig. 7. Visualization of instance and semantic similarity matrices. One row for each sample. From left to right, they are real scene blocks (each has two chairs), ground truth (instance), point cloud reflecting semantic similarity, point cloud reflecting instance similarity.

Table 8. Speed and Memory

Method	Backbone	Speed with/without clustering	GPU memory cost
Pointnet++	PointNet++	1859/**322**sec	4500MB
ASIS	PointNet++	2146/501sec	4500MB+64MB
Our model	PointNet++	**1649**/361sec	4500MB+64MB

6.3 Efficiency Study

In Tab. 8, we report the computation speed and memory cost of ours and some other methods. For memory cost, with size of 4096×4096 and single precision, our similarity matrix will cost $64M$ memory. Though we have two similarity matrices, they are constructed sequentially, so the maximum cost of GPU memory is $4500M+64M$. ASIS also has a matrix of size 4096×4096. The storage of the similarity matrix can be further reduced with one-way/three-way, criss-cross connection operations [49, 50].

For computation speed, ASIS is the slowest one, because it needs another KNN step. Though our method will spend more time on network feed-forward (without cluster op) than the backbone, we are faster over the whole process (with cluster op) because we divide the features of different instances far apart and make mean-shift converge quickly. In summary, our similarity matrices do not cost too much computation and memory.

7 Conclusion

We present Bi-Directional Attention Networks (BAN) for joint instance and semantic segmentation. Instead of element-wised fusing features for two tasks, our Bi-Directional Attention module builds instance and semantic similarity matrices from the instance and semantic features, respectively, with which two attention operations are conducted to bi-directionally aggregate features implicitly, introduce non-local information and avoid potential task conflict. Experiments on the three prevalent datasets S3DIS, PartNet and ScanNetV2 and method analysis suggest that the Bi-Directional Attention module could help give uniform and robust results within the same semantic or instance regions, and would also help to back-propagate uniform and robust gradients for optimization. Our BAN demonstrates significant superiority compared with baseline and other state-of-the-art works on the instance and semantic segmentation tasks consistently. Moreover, the ablation, mechanism and efficiency study further verifies the design and effectiveness of the Bi-Directional Attention module.

Acknowledgment

This work was funded by National Key Research & Development Plan of China (No. 2017YFB1002603), National Natural Science Foundation of China (61702301, 61772318) and Fundamental Research Funds of Shandong University.

References

1. Nguyen, A., Le, B.: 3d point cloud segmentation: A survey. In: 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM), IEEE (2013) 225–230
2. Ioannidou, A., Chatzilari, E., Nikolopoulos, S., Kompatsiaris, I.: Deep learning advances in computer vision with 3d data: A survey. ACM Computing Surveys (CSUR) **50** (2017) 1–38
3. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. (2015) 1990–1998
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2961–2969
5. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2359–2367
6. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3150–3158
7. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: European Conference on Computer Vision, Springer (2016) 534–549
8. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:1708.02551 (2017)
9. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 945–953
10. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5648–5656
11. Shi, B., Bai, S., Zhou, Z., Bai, X.: Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters **22** (2015) 2339–2343
12. Guerry, J., Boulch, A., Le Saux, B., Moras, J., Plyer, A., Filliat, D.: Snapnet-r: Consistent 3d multi-view semantic labeling for robotics. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2017) 669–678
13. Thanh Nguyen, D., Hua, B.S., Tran, K., Pham, Q.H., Yeung, S.K.: A field model for repairing 3d shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5676–5684
14. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1912–1920
15. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE (2015) 922–928
16. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3577–3586
17. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics (TOG) **36** (2017) 1–11
18. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 652–660

19. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. (2017) 5099–5108
20. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3d segmentation of point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2626–2635
21. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) **38** (2019) 1–12
22. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4558–4567
23. Hua, B.S., Tran, M.K., Yeung, S.K.: Pointwise convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 984–993
24. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems. (2018) 820–830
25. Ye, X., Li, J., Huang, H., Du, L., Zhang, X.: 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 403–417
26. Rethage, D., Wald, J., Sturm, J., Navab, N., Tombari, F.: Fully-convolutional point networks for large-scale point clouds. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 596–611
27. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9621–9630
28. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020) 5588–5597
29. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4421–4430
30. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3947–3956
31. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3d instance segmentation on point clouds. In: Advances in Neural Information Processing Systems. (2019) 6737–6746
32. Wang, X., Liu, S., Shen, X., Shen, C., Jia, J.: Associatively segmenting instances and semantics in point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4096–4105
33. Pham, Q.H., Nguyen, T., Hua, B.S., Roig, G., Yeung, S.K.: Jsis3d: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 8827–8836
34. Zhao, L., Tao, W.: Jsnet: Joint instance and semantic segmentation of 3d point clouds. In: Thirty-Fourth AAAI Conference on Artificial Intelligence. (2020) 12951–12958

35. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7794–7803
36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. (2017) 5998–6008
37. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 267–283
38. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3146–3154
39. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1534–1543
40. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 909–918
41. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5828–5839
42. Wang, W., Yu, R., Huang, Q., Neumann, U.: Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2569–2578
43. Elich, C., Engelmann, F., Schult, J., Kontogianni, T., Leibe, B.: 3d-bevis: birds-eye-view instance segmentation. arXiv preprint arXiv:1904.02199 (2019)
44. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE transactions on pattern analysis and machine intelligence **17** (1995) 790–799
45. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 6656–6664
46. Liu, S., Jia, J., Fidler, S., Urtasun, R.: Sgn: Sequential grouping networks for instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 3496–3504
47. Zhuo, W., Salzmann, M., He, X., Liu, M.: Indoor scene parsing with instance segmentation, semantic labeling and support relationship inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5429–5437
48. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) (2015)
49. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: Advances in Neural Information Processing Systems. (2017) 1520–1530
50. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 603–612