Instance-Aware Embedding for Point Cloud Instance Segmentation

Tong He, Yifan Liu, Chunhua Shen(✉), Xinlong Wang, and Changming Sun

The University of Adelaide, Adelaide, Australia
{tong.he,yifan.liu04,chunhua.shen,xinlong.wang}@adelaide.edu.au,
changming.sun@data61.csiro.au

Abstract. Although recent works have made significant progress in encoding meaningful context information for instance segmentation in 2D images, the works for 3D point cloud counterpart lag far behind. Conventional methods use radius search or other similar methods for aggregating local information. However, these methods are unaware of the instance context and fail to realize the boundary and geometric information of an instance, which are critical to separate adjacent objects. In this work, we study the influence of instance-aware knowledge by proposing an Instance-Aware Module (IAM). The proposed IAM learns discriminative instance embedding features in two-fold: (1) Instance contextual regions, covering the spatial extension of an instance, are implicitly learned and propagated in the decoding process. (2) Instance-dependent geometric knowledge is included in the embedding space, which is informative and critical to discriminate adjacent instances. Moreover, the proposed IAM is free from complicated and time-consuming operations, showing superiority in both accuracy and efficiency over the previous methods. To validate the effectiveness of our proposed method, comprehensive experiments have been conducted on three popular benchmarks for instance segmentation: ScannetV2, S3DIS, and PartNet and achieve state-of-the-art performance. The flexibility of our method allows it to handle both indoor scenes and CAD objects.

Keywords: 3D point cloud · Instance segmentation · Instance-aware

1 Introduction

The task of instance segmentation has recently gained popularity. As an extension to semantic segmentation, this task needs to separate pixels/points that have identical categories into individual groups. In the 2D image domain, many approaches [4,5,10,12,18] have been proposed and achieve promising results. With the growth of the availability of 3D sensors, more and more researches have focused on 3D scene understanding, which is a fundamental necessity for robotic vision, autonomous driving, and virtual reality. Although instance segmentation in the 3D domain has started to draw attention and has been discussed

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12375, pp. 255–270, 2020.
https://doi.org/10.1007/978-3-030-58577-8_16

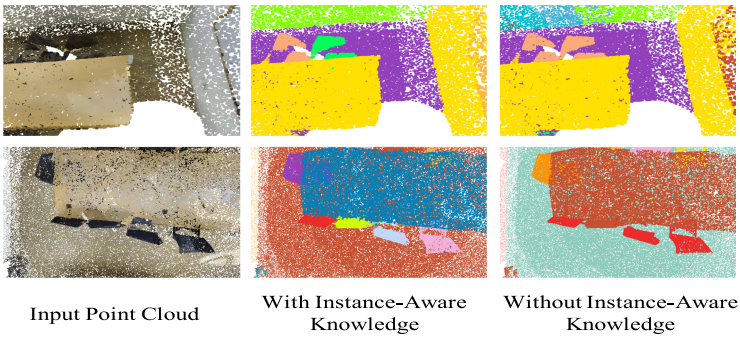


Fig. 1. Comparison of the instance segmentation results with and without the proposed Instance-Aware Module (IAM). The proposed IAM successfully encodes instance-aware information and geometric knowledge, which are critical for separating adjacent instances. Note that different instances can be presented in different colours. (Color figures online)

in [21,29,30,33,34], it still lags behind its 2D image counterpart and far from being solved.

Similar to the tasks of dense prediction in 2D images [2,16,35], context is also important in 3D domain. For 3D point clouds, PointNet++ [24] is the first work that captures local structure information and has been successfully utilized in the task of semantic segmentation. It maintains an encoder-decoder architecture, which includes several set-abstraction layers and feature-propagation layers for down-sampling and up-sampling, respectively. Algorithms such as radius search and k nearest neighbours (K-NN) search are utilized for aggregating local context knowledge. Building on this powerful network, many methods [21,29,30] have been proposed to tackle the task of instance segmentation on point clouds. To encode meaningful context information, ASIS [30] is proposed to associate two tasks together so they can cooperate with each other. JSIS3D [21] applied multi-value Conditional Random Field (CRF) that formulates a joint optimization for semantic segmentation and instance segmentation in a unified framework. However, these methods fail to explicitly encode the *instance contextual knowledge* and *geometric information*, which are extremely critical for separating adjacent instances and handling complex situations. For example, two neighbouring chairs can be easily confused and grouped as one united instance if boundaries and geometric information are not encoded in the embedding space (e.g., the second row in Fig. 1). In this paper, we address the problem by proposing an Instance-Aware Module (IAM) to learn the instance level context by locating representative regions for each input point. Moreover, geometric knowledge is explicitly encoded in the embedding space, which is an informative indicator to identify the points belonging to the same instance. The whole framework can be trained in an end-to-end manner to tackle instance segmentation and semantic segmentation simultaneously with little computation resource overhead.

Specifically, as shown in Fig. 2, our method maintains an encoder-decoder architecture. Different from previous methods that only maintain an instance grouping branch and a semantic segmentation branch, we come up with a novel light-weight instance-aware module, which localizes representative points within the same instance for each input point. The information from these representative points is then aggregated into the decoding process of the instance branch, generating instance-aware contexts for learning discriminative point-level embeddings. Moreover, the normalized geometric centroids of these representative points (predicted by every input point feature), are directly added to the embedding space, which provides critical geometric knowledge for identifying and reducing the ambiguity of adjacent instances.

The training of the instance-aware module is regularized jointly by the bounding box and instance segmentation supervision, such that the meaningful semantic regions can be tightly bonded by the spatial extension of the instance and guided towards representative regions of the instance.

Compared with the conventional representation of an instance by using vertexes to represent a bounding box, learning semantically meaningful regions helps to remove unrelated background and noise information. As it is applied in the bottleneck layer, very few additional computations are introduced. Compared with ASIS [30], which needs to search neighbours of every input point exhaustively, our approach shows superiority in both efficiency and effectiveness.

To validate the effectiveness of our proposed method, extensive experiments have been conducted on three popular benchmarks. The flexibility of our method allows it to be applied in not only indoor scenes but objects with fine-grained part labels. State-of-the-art performances are achieved on these datasets. To summarize, our main contributions are listed as follows.

- We propose a novel Instance-Aware Module, which successfully encodes instance-dependent context information for point cloud instance segmentation.
- Our method explicitly encodes instance-related geometric information, which is informative and helpful to produce discriminative embedding features.
- The proposed framework can be trained in an end-to-end manner and shows superiority over previous methods on both efficiency and effectiveness. With the proposed method, state-of-the-art results are achieved on different tasks.

2 Related Works

Instance segmentation on point clouds has just started to be discussed recently. In this section, we briefly review some existing approaches that are related to this field.

2.1 Deep Learning on Point Clouds

Deep learning-based methods for 3D feature extraction can be roughly categorized into three classes: voxel-based, multi-view-based, and point-based. Voxel-

based methods [9,19,25,32] utilize 3D convolution neural networks for feature extraction on voxelized spatial grids, which can be easily influenced by the density of the points. Meanwhile, it is highly constrained by the huge memory occupation and lower running speed because a large proportion of computation is wasted on vacant voxels. Many approaches have been proposed to address the problem [9,25]. Octree [25] tries to modify the convolution operation by generating average hidden states in empty space. SparseConv [9] is proposed to process spatially sparse data more efficiently by encoding with a Hash Table to avoid unnecessary memory usage in vacant space. The second category is multi-view-based methods [13,23,26], which first project 3D shapes or point clouds into 2D images and utilize conventional 2D CNN for feature extraction. Hou *et al.* proposed 3D-SIS [13] by leveraging both RGB 2D input and 3D geometrical information. 2D features are then back-projected into 3D grids. Unlike the above methods, directly extracting features on point clouds is more efficient and straightforward. PointNet [22] is the pioneering work that directly learns a spatial encoding of each point. A symmetrical function is utilized to process disordered point sets. To effectively encode local context information to obtain representative features, many approaches [14,15,24,27,28] have been proposed. Qi *et al.* proposed PointNet++ [24] which applied PointNet recursively on a nested partitioning of the input point clouds. Thomas *et al.* came up with KPConv [27] by designing a continuous weight space through interpolating with several kernel points. In our experiments, we utilize PointNet++ as the backbone to verify the effectiveness of our method.

2.2 Instance Segmentation on Point Cloud

Although the task of instance segmentation on 2D images has made huge progress since Mask-RCNN [10] was proposed, its 3D point cloud counterpart lags far behind. SGPN [29] is the first deep-learning-based method developed in this field. It tried to generate point cloud groups by predicting three objectives: the similarity matrix, the confidence map, and the semantic prediction map. Due to the pair-wise term, the method occupies a large amount of GPU memories and suffers from slow running speed and small batch size for training. On the other hand, generating instance groups from three matrices requires many hyper-parameters, making it less stable for different scenarios. Wang *et al.* proposed ASIS [30] to address the problem by removing the pair-wise prediction and introducing a discriminative loss for instance embedding. The loss pulls the embeddings of the same instance towards the cluster center and pushes the cluster centers away from each other. However, the method fails to utilize the geometrical information and is unaware of the spatial distribution of the instances. GSPN [34], proposed by Yi *et al.*, generates shape proposals using a generative model for instance segmentation. Due to its emphasis on geometric understanding for object proposal, it achieved promising performance on both indoor dataset and part instances dataset. Due to the large requirement of GPU memory and a two-step training procedure, it is ineffective with limited computation resources. MPNet [11] proposed a memory-based module to deal with the

imbalance of the point cloud data. In this work, we propose an Instance-Aware Module (IAM) to encode instance context knowledge and geometric information. The state-of-the-art performance on three large open benchmarks shows superiority over previous methods in both effectiveness and efficiency.

3 Method

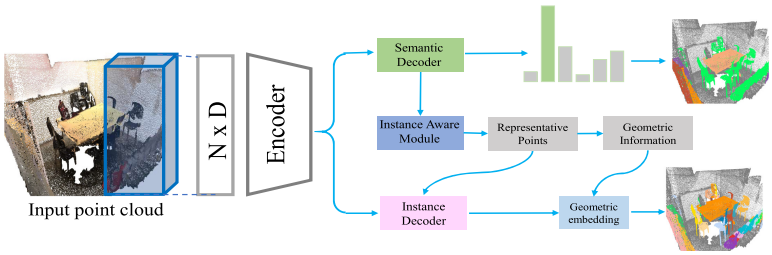


Fig. 2. The whole framework of our proposed one-stage method, which is a simple and clear encoder-decoder architecture. The input point clouds first go through a shared encoder network, and two parallel decoders are followed: one for semantic segmentation, one for instance grouping. A novel instance aware module (IAM) is proposed to generate representative points for instance segmentation. We use the coordinates of representative points to select argument features for instance segmentation module and the geometric information of the coordinates to extend the instance embedding. The whole framework is end-to-end trainable.

In this section, we describe our proposed Instance-Aware Module (IAM), which can encode both instance-aware context and instance-related geometric information. Details of the approach are presented below.

3.1 Network Framework

As shown in Fig. 2, we apply an encoder-decoder architecture. The encoder is shared by two tasks and takes point sets $P \in \mathbb{R}^{N \times D}$ as input, where N denotes the total number of the points and D refers to the input feature dimension. The input features can consist of colour and position information, e.g., X, Y, Z, R, G, and B. The decoder contains two parallel branches: one for semantic segmentation, one for instance embedding. The semantic segmentation branch generates per-point classification results $S \in \mathbb{R}^{N \times D_c}$, where D_c is the category number. Focal loss [17] L_{fl} is applied to address the category imbalance during the training process. Besides, the instance branch outputs per-point embedding features $E \in \mathbb{R}^{N \times D_e}$ for learning a distance metric, where D_e is the embedding dimension. The embeddings belonging to the same instance should end up close together, and the embeddings belonging to the different instances should end up

far apart. During the inference, a clustering algorithm is applied to obtain the final grouping results. A novel IAM for producing instance aware knowledge is achieved by detecting the spatial extension of an instance. Through IAM, representative points locating on the corresponding instance provide instance-aware knowledge, which contains two parts: (1) instance-related contextual information via detection a set of regions that are tightly covering the spatial extension of an instance. (2) instance geometric knowledge that is critical for separating adjacent objects.

3.2 Instance-Aware Module

We propose an instance-aware module (IAM) mainly for selecting representative points that capture spatial instance context. For point p_i with position x_i, y_i and z_i, point-level offsets are predicted by the contextual detection branch to represent the spatial extension of the instance, denoted as $\{\Delta x_i^k, \Delta y_i^k, \Delta z_i^k\}_{k=1}^K$. Representative regions of the instance predicted by p_i is $\mathcal{R}_i$, which can be simply represented as:

$$\mathcal{R}_i = \{(x_i + \Delta x_i^k, y_i + \Delta y_i^k, z_i + \Delta z_i^k)\}_{k=1}^K, \tag{1}$$

where K is the number of representative points and i represent the i-th point. The axis-aligned bounding box predicted by every point can be formulated as $\mathcal{B}_i$ through a min-max function F: $\mathcal{B}_i = F(\mathcal{R}_i)$

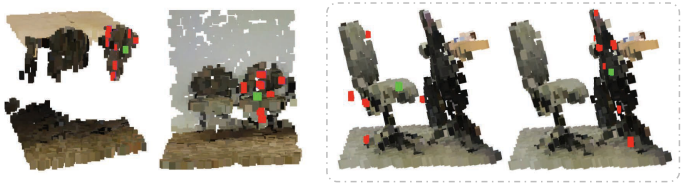
Learning these representative regions is jointly driven by both the spatial bounding boxes and the instance grouping labels, such that $\mathcal{R}_i$ can tightly compass the instance. To achieve this, three losses are provided: L_{bnd}, L_{cen} and L_{ins} (the last two will be discussed in the next section). L_{bnd} is to maximize the overlaps of the bounding boxes between the prediction and the ground truth. 3D IoU loss is utilized in our paper:

$$L_{bnd} = \frac{1}{N} \sum_{i=1}^{N} 1 - IoU(GT_i, \mathcal{B}_i), \tag{2}$$

where N is the total number of points, $\mathcal{B}_i$ is the predicted bounding box of the i-th point, and GT_i is the 3D axis-aligned bounding box ground truth of the i-th point. To have a better understanding of the detection branch, we visualize $\mathcal{R}_i$ in Fig. 3. Green points are selected p_i, and red points are the predicted $\mathcal{R}_i$. We choose the number of representative points as 18, which empirically works well in our experiments. Employing more points will have limited improvements. Therefore, in terms of efficiency, we choose $K = 18$. Instance related regions are located and successfully cover the spatial extension. In the next section, we provide details of how to incorporate these instance contextual information.

3.3 Instance Branch

Conventionally, the inputs of the instance decoder are down-sampled bottleneck points $P_b \subseteq P$, and the corresponding features are denoted as F_b. These features



(a) (b) (c)

Fig. 3. Visualization of detected representative points. The green point is randomly selected, and red points are the corresponding meaningful regions output by the IAM. Due to the encoded instance context information, our method can separate adjacent objects. (Figure best viewed in color) (Color figures online)

are gradually propagated to the full set of points through several up-sampling layers. To encode the instance context during the propagation process, we utilize the meaningful semantic regions of $\mathcal{R}_b$ for the bottleneck points.

Encode Instance-Aware Context. Representations of F_b are augmented by aggregating information from $\mathcal{R}_b$ that covers the instance spatial extent. As these detected points are not necessarily located on the input points, the features of $\mathcal{R}_b$ are interpolated by using K-NN. The interpolated features are then added to the original F_b, generating features containing both local representation and instance context. Compared with ASIS [30], which has to search neighbours for every input point, our method, on the other hand, is more efficient. As K-NN is applied in the bottleneck layer, the searching space in P_b is much smaller than that in P, introducing very limited computation overhead. The combined features are gradually upsampled during the decoding process, propagating the instance-aware context through all points.

Encode Geometric Information. Geometric information is critical for identifying two close objects. To learn a discriminative embedding feature, we directly concatenate the normalized centroids of coordinates to the embedding space. Considering the centroid $C(\mathcal{B}_i)$ predicted by point p_i, where $C(\cdot)$ is the function for computing geometric centroids of a given bounding box, the final per-point embedding feature can be represented as $\hat{E}_i = \text{Concat}(E_i, C(\mathcal{B}_i))$, where E_i is the embedding feature produced from the instance branch. Besides, to force the geometric information to be consistent for the points that have identical instance label, we pull the predicted geometric centroids from the same instance towards the cluster center by:

$$L_{cen} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N_m}\sum_{i=1}^{N_m}[\|C(\mathcal{B}_i) - \mu_m\| - \sigma_v]_+^2, \tag{3}$$

where M is the total number of instances, and N_m is the point number for m-th instance. μ_m refers to the average predicted geometric centroids of m-th

instance. $[x]_+$ is defined as$[x]_+ = \max(0, x)$ and σ_v is the loose margin. The L_{cen} is designed for forcing the additional geometric information to have less variation and to be informative for separating adjacent objects.

The informative per-point embedding $\{\hat{E}\}_{n=1}^{N}$ is applied for learning a distance metric that could pull intra-instance embedding toward the cluster center and push instances centers away from each other. The loss function is formulated as:

$$\begin{aligned} L_{ins} = & \underbrace{\frac{1}{M(M-1)} \sum_{a=1}^{M} \sum_{\substack{b=1 \\ b \neq a}}^{M} [2\sigma_d - \|\mu_a - \mu_b\|]_+^2}_{inter-instance} \\ & + \underbrace{\frac{1}{M} \sum_{i=1}^{M} \frac{1}{N_m} \sum_{m=1}^{N_m} [\|\mu_m - \hat{E}_m\| - \sigma_v]_+^2}_{intra-instance}, \end{aligned} \tag{4}$$

where M is the total instance number, N_m is the point number of the m-th instance. σ_d and σ_v are relaxation margins. During the training process, the first term pushes instance clusters away from each other and the second term pulls the embedding towards the cluster center. During the inference process, a fast mean-shift algorithm is applied for clustering different instances in the embedding spaces.

To summarize, our method is end-to-end trainable and supervised by four losses. The loss weights for the four losses are all set to 1 in all our experiments.

$$L = L_{fl} + L_{bnd} + L_{cen} + L_{ins}, \tag{5}$$

4 Experiments

In this section, we evaluate the effectiveness of our proposed method. Both qualitative and quantitative experiments are conducted and reported.

4.1 Datasets

We introduce three popular datasets that have instance annotations: Stanford 3D Indoor Semantic Dataset (S3DIS) [1], ScanNetV2 [3], and PartNet [20]. S3DIS is collected in 6 large-scale indoor areas, covering 272 rooms. The whole dataset contains more than 215 million points and is consisted of 13 common semantic categories. ScanNetV2 [3] is an RGB-D video dataset. It contains more than 1500 scans, which is split into 1201, 300, and 100 scans for training, validation, and testing, respectively. The dataset contains 40 classes in total, and 13 categories are evaluated. Different from the above two datasets, PartNet [20] is a consistent large-scale dataset with fine-grained object annotations. It consists of more than 570k part instances covering 24 object categories. Each object contains 10000 points. Similar to GSPN [34], we select five categories that have the largest number of training examples.

4.2 Evaluation Metrics

On the S3DIS dataset, we conduct 6-fold cross-validation. Similar to SGPN [29] and ASIS [30], the performance on Area-5 is also reported. On ScanNetV2 [3], we report our results on the validation set, which contains more instances and has more stable results. On the PartNet [20] dataset, five selected categories are Chair, Storage, Table, Lamp, and Vase. Both coarse and fine-grained results are included. Different levels of different categories are trained separately and independently. The evaluation metrics for semantic segmentation are the overall pixel-wise accuracy ($mAcc$), category-wise mean accuracy ($oAcc$) and average intersection-over-union ($mIoU$). The instance segmentation is evaluated by the average instance-wise coverage ($mCov$), mean weighted instance-wise coverage ($mWCov$), mean instance precision ($mPrec$) and recall ($mRec$) with IoU threshold of 0.5. The weights for $mWCov$ is calculated by $w_i = \frac{|N_i|}{\sum_k |N_k|}$, where i is the i-th instance and N_k is the point number of k-th ground truth instance.

4.3 Implementation Details

For the S3DIS [1] and ScanNetV2 [3], each scan contains millions of points, making it hard to process all data at one time. In our experiments, we split each scene into $1m \times 1m$ overlapped blocks with 0.5 m stride. Then, 4,096 points are randomly sampled across each block. Similar to SGPN [29], every point is represented by a 9-D feature (X, Y, Z, R, G, B, and normalized positions in blocks N_X, N_Y, N_Z). PartNet [20], on the other hand, is proposed for shape analysis which contains 10,000 points for each instance. We randomly select 8,000 for training and 10,000 for testing.

Although our method is not restricted to any specific network, all experiments are conducted with vanilla PointNet++ [24] as the backbone (without multi-scale grouping) and leave the other choices for future study. One single GTX1080Ti GPU card is used for training with the batch size set to 16. The initial learning rate is set to 0.01 (0.001 for S3DIS) and divided by 2 in every 300k iterations. We use Adam optimizer with momentum set to 0.9, and the whole network is trained for 100 epochs. The hyper-parameters for discriminative loss are identical with original setting in [30]: $\sigma_v = 0.5$, $\sigma_d = 1.5$. Besides, for testing the whole scene on S3DIS and ScanNetV2, a method named BlockMerging [29] is used for grouping blocks according to the segmentation information of the overlapped areas.

4.4 Ablation Studies

We first build a strong baseline that contains two decoder branches: one is the semantic segmentation, and the other is the instance embedding branch. Two losses are used for supervising the two branches: the cross-entropy loss for the segmentation task and the discriminative loss for instance grouping. The discriminative loss forces points belonging to the same instance to lie close together in the embedding space and keep a large margin for points belonging to different instances. The loss weights are set to 1.0. We conduct our experiments on the ScanNetV2 validation set.

Table 1. Ablation study on ScanNetV2 dataset. Both AP_{50} and AP_{25} are reported on the validation set. **FL** refers to focal loss. **InsContext** refers to instance-aware context. $\mathbf{L_{cen}}$ refers to centroid constrain loss in Eq. 3. **GE** refers to geometric embedding.

Method	FL	InsContext	L_{cen}	GE	AP_{50}	AP_{25}
Baseline					22.0	45.2
	✓				24.0	45.5
	✓	✓			27.6	48.2
	✓	✓	✓		28.9	48.9
Ours	✓	✓	✓	✓	**31.5**	**50.4**

Focal Loss. Focal loss [17] is first proposed in the object detection task to address the problem of data imbalance between positive and negative samples. Due to the imbalance of categories introduced in the point cloud, we apply focal loss in the segmentation branch with default parameters identical to [17]. The results are shown in Table 1, and the focal loss can improve the results by 2.0 for AP_{50}, from 22.0 to 24.0.

Instance Aware Module. We study the influence of the proposed instance-aware module, which first finds out representative points of the instance, and then features from these sampled points are aggregated. Encoding the spatial extension knowledge helps to separate and distinguish close instances. As shown in Table 1, the instance aware decoder boosts the performance by a large margin, improving AP_{50} from 24.0 to 27.6 and AP_{25} from 45.5 to 48.2. Besides, simply enlarging the dimension of the embedding space can not bring further improvement in performance (presented in ASIS [30]). The proposed geometric embedding provides informative knowledge, which brings about 2.6% improvement in AP_{50}, demonstrating the effectiveness of our proposed method. Qualitative results are shown in Fig. 4. Our method shows robustness to the intensive scenes, which require more discriminative features to separate different instances.

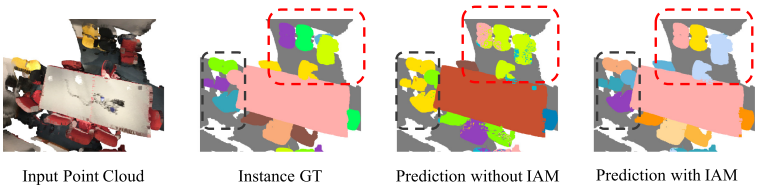


Fig. 4. Comparison of the results with and without the Instance-Aware Module. Due to the successfully encoded instance context and geometric information, our method generates discriminative results, especially for nearby objects.

Table 2. Instance segmentation results on the S3DIS dataset. Both Area-5 and 6-fold performance are reported. **mCov**: mean instance-wise IoU coverage. **mWCov**: mean size-weighted IoU coverage. **mPrec**: mean precision with IoU threshold 0.5. **mRec**: mean recall with IoU threshold of 0.5. All our results are achieved on a vanilla PointNet++ [24] backbone without multi-scale grouping for fair comparison.

Method	Year	mCov	mWCov	mPrec	mRec
Test on Area 5					
SGPN [29]	2018	32.7	35.5	36.0	28.7
ASIS [30]	2019	44.6	47.8	55.3	42.4
3D-BoNet [33]	2019	-	-	57.5	40.2
JSNet [36]	2020	48.7	51.5	**62.1**	46.9
Ours	-	**49.9**	**53.2**	61.3	**48.5**
Test on 6-fold					
SGPN [29]	2018	37.9	40.8	31.2	38.2
MT-PNet [21]	2019	-	-	24.9	-
MV-CRF [21]	2019	-	-	36.3	-
ASIS [30]	2019	51.2	55.1	63.6	47.5
3D-BoNet [33]	2019	-	-	65.6	47.6
PartNet [20]	2019	-	-	56.4	43.4
Ours	-	**54.5**	**58.0**	**67.2**	**51.8**

Table 3. Comparison per-class performance of our proposed method with state-of-the-arts on the S3DIS semantic segmentation task, tested on all areas (6-fold). Our result utilize the vanilla PointNet++ [24] without multi-scale group. Even with a simple baseline, the proposed method surpassed the complex graph-based methods. **mA:** mean pixel-wise accuracy. **mI:** mean category-wise IoU.

	mA	mI	cei.	flo.	wall	beam	col.	win.	door	tab.	cha.	sofa	boo.	boa.	clu.
[22]	78.5	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
[7]	79.2	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
[7]	81.1	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
[24]	-	53.2	90.2	91.7	73.1	42.7	21.2	49.7	42.3	62.7	59.0	19.6	45.8	48.2	45.6
[8]	-	58.3	92.1	90.4	**78.5**	37.8	35.7	51.2	65.4	64.0	**61.6**	25.6	51.6	49.9	53.7
[31]	84.1	56.1	-	-	-	-	-	-	-	-	-	-	-	-	-
[14]	85.9	60.0	93.1	95.3	78.2	33.9	**37.4**	**56.1**	**68.2**	64.9	61.0	34.6	51.5	51.1	**54.4**
Ours	**86.5**	**60.2**	**94.0**	94.1	76.6	**53.4**	33.6	54.2	62.7	**70.2**	60.2	**36.6**	**53.4**	**54.3**	53.5

Centroid Constrain Loss. The centroid constraint loss L_{cen} is designed for maintaining consistency for points belonging to the same instance. The loss function serves as a regularizer to constrain the embedding features from the same instance to have a small variance. Moreover, it also helps stabilize the centroids when concatenated to the embedding space. As can be inferred from

Table 1, the utilization of L_{cen} improves the AP_{50} from 27.6 to 28.9. By further combing the geometric embeddings with the per-point features, we achieve an improvement on the AP_{50} from 28.9 to 31.5.

Table 4. Instance segmentation results on ScanNetV2 benchmark (validation set). The metric of mAP@0.25 is reported. All methods except [8] are based on PointNet or PointNet++. (Categories of Table, Toilet, and Window are not presented in the table.)

	mAP	bat	bed	she	cab	cha	cou	cur	des	doo	oth	pic	ref	shc	sin	sof
[10]	26.1	33.3	0.2	0.0	5.3	0.2	0.2	2.1	0.0	4.5	2.4	23.8	6.5	0.0	1.4	10.7
[6]	-	66.7	56.6	7.6	3.5	39.4	2.7	3.5	9.8	9.9	3.0	2.5	9.8	37.5	12.6	60.4
[29]	35.1	20.8	39.0	16.9	6.5	27.5	2.9	6.9	0.0	8.7	4.3	1.4	2.7	0.0	11.2	35.1
[30]	47.4	57.3	52.1	1.4	18.5	46.1	19.2	20.3	13.3	13.8	18.8	6.6	17.6	33.1	8.8	32.1
Ours	50.4	63.0	60.9	0.2	22.9	67.2	10.2	18.6	10.5	15.5	22.7	9.5	16.5	55.2	13.6	34.3

4.5 Comparison with State-of-the-Art Methods

In this section, we make a comprehensive comparison with other state-of-the-art methods on three popular benchmarks. Our method can not only be applied to indoor scenes but also achieved promising results on the hierarchical 3D part dataset. The results on S3DIS [1], ScanNetV2 [3], and PartNet [20] show the superiority of our method on both efficiency and effectiveness.

Training and Testing Efficiency. As the first method to solve instance segmentation on the point cloud, SGPN [29] needs to predict a pair-wise similarity matrix, which requires a lot of memory. Each sample requires about 2.7G for training. GSPN [34] needs two training stages, and each sample has to take about 6G memory for training due to the generative network. ASIS [30] addresses the problem by removing the memory consuming parts and learning a discriminative embedding. However, due to the massive usage of K-NN for every point, training ASIS requires a memory of more than 700M for every sample and the inference time for the network requires 60 ms for each block. As we only utilize K-NN in the bottleneck layer, training IAM needs only about 400M for each sample and reduces the running time to 42 ms for each block, showing the superiority in both the effectiveness and efficiency of our method.

Quantitative Results on S3DIS. Instance segmentation performance on Area-5 and k-fold cross validation results are reported in Table 2. We compare our method with other start-of-the-art results. Equipped with instance-aware knowledge, 2.4%, and 7.7% improvement are achieved with metric $mPrec$ and $mRec$ for instance segmentation. Although employing a simple backbone, our

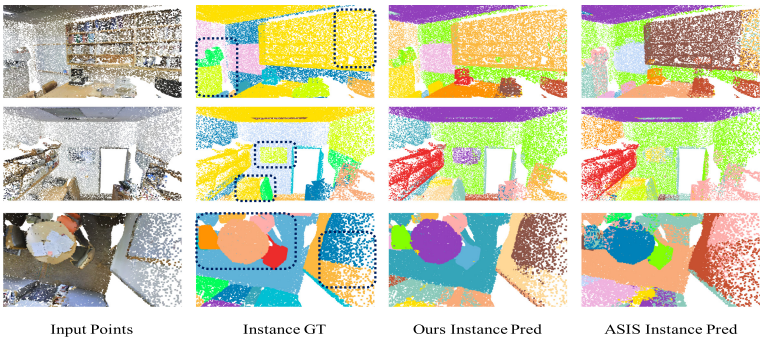


Fig. 5. Visualization of the instance segmentation results on the S3DIS indoor scenes. From left to right are: input point cloud, the ground truth of instance segmentation, the results of our proposed method, and the results of ASIS [30]. As shown in the figure, our methods have discriminative embedding features for distinguishing adjacent objects. We should note that: different instances are presented with different colors, and the same instances in different methods are not necessarily sharing the same color. (Color figure online)

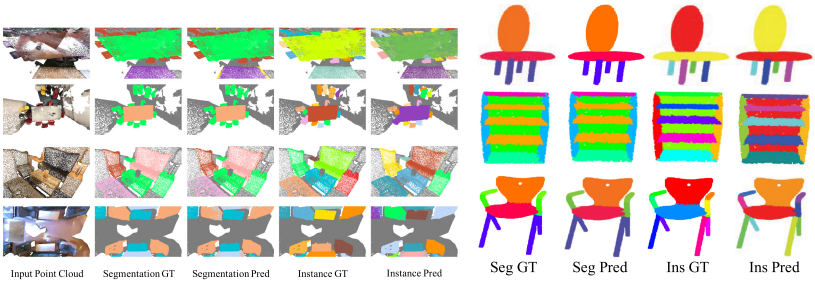
method surpasses previous methods, which need more complex operations and more memories for training. Moreover, we also report the performance on the semantic segmentation task in Table 3. The results are evaluated with 6-fold cross-validation. Our method is built upon vanilla PointNet++ [24] and achieves better results compared with methods that applied multi-view [7] or even graph CNN [14,31]. Qualitative instance grouping results are shown in Fig. 5. We compare the performance of our method with ASIS [30], showing the effectiveness of the encoded instance-aware knowledge.

Table 5. Instance segmentation results on PartNet. We report part-category mAP (%) under IoU threshold 0.5. There are three different levels for evaluation: coarse-grained level, middle-grained level, and fine-grained level. We select five categories with the most data amount for training and evaluation.

	Level1					Level 2					Level 3				
	Cha	Sto	Tab	Lam	Vas	Cha	Sto	Tab	Lam	Vas	Cha	Sto	Tab	Lam	Vas
[29]	72.4	32.9	49.2	32.7	46.6	25.4	30.5	18.9	21.7	-	19.4	21.5	14.6	14.4	36.5
[20]	74.4	**45.2**	54.2	**37.2**	49.8	35.5	35.0	31.0	26.9	-	29.0	27.5	23.9	18.7	52.0
[34]	-	-	-	-	-	-	-	-	-	-	26.8	26.7	21.9	18.3	-
[34]	77.1	43.2	55.0	34.1	48.5	36.0	35.5	31.3	24.8	-	26.8	26.7	21.9	18.3	51.9
Ours	**79.5**	44.2	**56.1**	36.1	**49.9**	**38.6**	**37.1**	**33.0**	**26.9**	-	**31.2**	**28.9**	**25.5**	**19.4**	**53.1**

Quantitative Results on ScanNetV2. The quantitative performance on ScanNetV2 is presented in Table 4. It is evaluated on the validation set. Both mAP@0.25 and mAP@0.5 are reported. The results of [30] and [34] are reproduced via the open source code. For fair comparison, methods based on PointNet [22] or PointNet++ [24] are reported. Compared with state-of-the-art ASIS [30], our method achieves promising results and boosts mAP@0.25 and mAP@0.5 with a significant improvement, by 8.4% and 6.5%, respectively. Figure 6(a) shows qualitative results of instance segmentation on ScanNetV2.

Quantitative Results on PartNet. The performance on PartNet [20] is shown in Table 5. Different from indoor scenes, PartNet provides fine-grained and hierarchical object parts annotations. Level-1 contains the coarsest annotations and level-3 contains the finest annotations. Similar to GSPN [34], we report the performance of the five categories that have the largest number of training samples: Chair, Storage, Table, Lamp, and Vase. mAP@0.5 is reported. Each category of different levels is trained separately. Our method achieved state-of-the-art results on most categories and levels, substantially improving the performance. Figure 6(b) shows qualitative results of instance segmentation on PartNet. Different categories and fine-grained levels are provided.



(a) Visualization of ScanNetV2 (b) Visualization of Partnet

Fig. 6. Visualization of the instance segmentation results on (a) ScanNetV2 and (b) Partnet. Our method successfully discriminates adjacent objects that are difficult to separate. Noting: different instances are presented with different colors, and the same instance in different methods are not necessarily sharing the same color. (Color figure online)

5 Conclusion

In this paper, we present a novel method for solving point cloud instance segmentation and semantic segmentation simultaneously. An instance-aware module (IAM) is proposed to encode both instance-aware context and geometric information. Extensive experimental results show that our method has achieved state-of-the-art performance on several benchmarks and shown superiority in both effectiveness and efficiency.

References

1. Armeni, I., et al.: 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (2018)
3. 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)
4. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the European Conference on Computer Vision (2016)
5. Dai, J., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
6. Elich, C., Engelmann, F., Kontogianni, T., Leibe, B.: 3D-BEVIS: Bird's-Eye-View Instance Segmentation. arXiv preprint arXiv:1904.02199 (2019)
7. Engelmann, F., Kontogianni, T., Hermans, A., Leibe, B.: Exploring spatial context for 3D semantic segmentation of point clouds. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2017)
8. Engelmann, F., Kontogianni, T., Schult, J., Leibe, B.: Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. arXiv:1810.01151 (2018)
9. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
11. He, T., Gong, D., Tian, Z., Shen, C.: Learning and Memorizing Representative Prototypes for 3D Point Cloud Semantic and Instance Segmentation. arXiv preprint arXiv:2001.01349 (2020)
12. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
13. 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)
14. Li, G., Müller, M., Thabet, A., Ghanem, B.: DeepGCNs: can GCNs go as deep as CNNs? In: Proceedings of the IEEE International Conference on Computer Vision (2019)
15. 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)
16. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
18. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

19. Maturana, D., Scherer, S.: VoxNet: a 3D convolutional neural network for real-time object recognition. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (2015)
20. Mo, K., et al.: 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)
21. Pham, Q.H., Nguyen, D.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)
22. 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)
23. 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)
24. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: Proceedings of Advances in Neural Information Processing Systems (2017)
25. Riegler, G., Ulusoy, A.O., Geiger, A.: OctNet: Learning Deep 3D Representations at High Resolutions. arXiv preprint arXiv:1611.05009 (2016)
26. 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)
27. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: KPConv: flexible and deformable convolution for point clouds. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
28. Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J.: Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
29. 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)
30. 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)
31. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**, 1–12 (2019)
32. Wu, Z., et al.: 3D ShapeNets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
33. Yang, B., et al.: Learning object bounding boxes for 3D instance segmentation on point clouds. In: Proceedings of the Advances in Neural Information Processing Systems (2019)
34. 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 (2018)
35. Zhang, H., et al.: Context encoding for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
36. Zhao, L., Tao, W.: JSNet: joint instance and semantic segmentation of 3D point clouds. In: Proceedings of AAAI Conference on Artificial Intelligence (2019)