IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

3D Instance Embedding Learning with a Structure-Aware Loss Function for Point Cloud Segmentation

Zhidong Liang[1], Ming Yang[2], Hao Li[2] and Chunxiang Wang[2]

***Abstract*— This paper presents a framework for 3D instance segmentation on point clouds. A 3D convolutional neural network is used as the backbone to generate semantic predictions and instance embeddings simultaneously. In addition to the embedding information, point clouds also provide 3D geometric information which reflects the relation between points. Considering both types of information, the structure-aware loss function is proposed to achieve discriminative embeddings for each 3D instance. To eliminate the quantization error caused by 3D voxel, the attention-based k-nearest neighbor (kNN) is proposed. Different from the average strategy, it learns different weights for different neighbors to aggregate and update the instance embeddings. Our network can be trained in an end-to-end style. Experiments show that our approach achieves state-of-the-art performance on two challenging datasets for instance segmentation.**

I. Introduction

With the development of 3D sensors such as RGB-D cameras, 3D scene understanding becomes more and more important in augmented reality, autonomous driving and robotics. Compared with 2D scene understanding, 3D understanding is more challenging due to the data sparsity and high computational cost. However, 3D data contain rich geometric information which is useful for semantic understanding whereas 2D images do not directly reflect such information. 3D understanding includes many tasks. In comparison to 3D semantic segmentation and object detection, 3D instance segmentation is more challenging since it simultaneously provides the semantic category and the instance identification. In this paper, we focus on 3D instance segmentation.

Instance segmentation in 2D images has achieved a great performance. Most approaches to 2D instance segmentation are proposal-based which first apply a proposal generator to obtain the initial region proposals [1]–[6] and then use a binary segmentation network to obtain the instance mask. Such idea [7] achieves desirable results thanks to accurate region proposals. However, these methods have some drawbacks. First, they are the combination of object detection and semantic segmentation. The training process is usually two-stage which is more complex than those of single-stage instance segmentation methods. Second, one pixel may have more than one instance label since it may be in two overlapped bounding boxes simultaneously. The second problem can be more serious when it is the multi-class instance segmentation of clutter scenes.

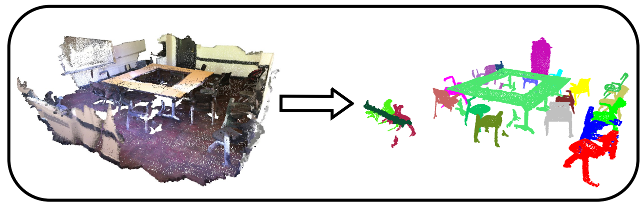

Fig. 1. The network takes RGB-D point clouds as input and outputs instance labels. The wall and the floor are ignored in the instance segmentation of the ScanNet dataset [11].

An alternative idea is to generate embeddings for each pixel [8]–[10] and then apply a clustering algorithm to obtain the final instance result. This idea utilizes semantic segmentation networks to generate discriminative embeddings. Although such proposal-free methods [8]–[10] can not get as high performance as proposal-based methods [7] on 2D images, they are simpler in the implementation and can avoid the drawbacks of proposal-based methods. Additionally, such a framework can simultaneously segment images at the semantic level and instance level while the proposal-based methods can only obtain the instance result. In this paper, we propose a proposal-free framework for 3D instance segmentation. Fig. 1 shows the input and output of our method.

SGPN [12] is a pioneering work on 3D instance segmentation. It proposes a loss function based on the similarity matrix to supervise the instance embedding. However, the complexity of its loss function is $O(n^2)$ where n is the number of points which makes it difficult to process large scenes on the order of 10^5 or more points. Inspired by the central idea in [13], we propose a structure-aware loss function that reduces the complexity to $O(n)$ and utilizes the structure information of point clouds. Previous research [13] only pays attention to the embedding center. In addition to the embedding information, point clouds also have 3D geometric information which is represented as the 3D coordinates. Therefore, we also present a geometric center for

Manuscript received February 21, 2020; accepted June 19, 2020. This paper was recommended for publication by Associate Editor Rudolph Triebel and Editor Tamim Asfour upon evaluation of the reviewers' comments. This work was supported in part by National Natural Science Foundation of China under Grants U1764264 and 61873165, and in part by Shanghai Automotive Industry Science and Technology Development Foundation (1807). *(Corresponding author: Ming Yang)*

[1]Zhidong Liang is with Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China (e-mail: lzd950512@sjtu.edu.cn).

[2]Ming Yang, Hao Li and Chunxiang Wang are with Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200031, China (e-mail: MingYang@sjtu.edu.cn; haoli@sjtu.edu.cn; wangcx@sjtu.edu.cn).

Digital Object Identifier 10.1109/LRA.2020.3004802

each instance in the loss function. Within an object instance, the embeddings of points near the geometric center of the object are more likely to be similar, whereas the embeddings of points near the edge are more likely to be different. This means that the embeddings near the edge are hard examples. To deal with this problem, the proposed loss function focuses more on points far from the geometric center. For instance segmentation, the local consistency of embeddings is very important. We can search the k-nearest neighbors (kNN) for each point according to their 3D coordinates and then average neighboring embeddings from neighbors to enhance the consistency. However, kNN is not so accurate for modeling the relation between two points. This means that a point and its neighbor may belong to different instances even if they are very close in the physical world. In this case, wrong information will be passed to the point when using a simple averaging strategy. Considering this problem, we propose the attention-based kNN. When aggregating information from neighbors, the weight for each neighbor is not fixed but learnable. Additionally, if using a 3DCNN as the backbone to extract features, then a quantization error will be caused by 3D voxel. kNN is a pointwise operation that can eliminate such an error.

In conclusion, the main contributions of this paper are as follows:

- We propose the structure-aware loss function to take both geometric information and embedding information into account for supervising instance embeddings.
- We propose the attention-based kNN to aggregate information from neighbors and eliminate the quantization error caused by 3D voxel.
- Experiments show that our proposed method achieves state-of-the-art performance on the ScanNet benchmark [11] and the NYUv2 dataset [14].

II. RELATED WORK

A. Instance Segmentation

Instance Segmentation on 2D Images. As a combination of object detection [1]–[6] and semantic segmentation [15]–[17], instance segmentation [7]–[10], [18] becomes a hot research topic since it provides richer semantic information. There are two main ideas for instance segmentation. One idea is proposal-based which is relevant to general object detection. [7] is an effective system for instance segmentation which segments the proposal bounding box to obtain the instance mask. Based on [7], [18] enhances the information propagation in representative pipelines and proposes a more flexible feature pooling method. The other idea is to learn an embedding for each pixel and then group pixels to form instance masks. [8] uses the Euclidean distance with a sigmoid function to measure the similarity of each pair of embedding vectors. [9] proposes a loss function to pull pixels belonging to the same instance closer in the embedding space. [10] utilizes the cosine similarity which is invariant to the scale of the embedding vector.

Instance Segmentation on Point Clouds. Recently, several researchers have tried instance segmentation on point clouds. [12] is a pioneer in 3D instance segmentation which generates an embedding for each point and proposes a double-hinge loss to supervise the embedding learning. [19] generates a proposal for each object by reconstructing the shape and then applies PointNet++ to obtain the final instance segmentation result. [20] proposes a detection-based method to obtain instance predictions by fusing multi-modal inputs. [21] proposes a detection-by-segmentation network for part instance segmentation. [22] predicts pixel-wise panoptic labels for 2D images using [7], [17] and then integrates the predicted 2D labels into a 3D volumetric map. [23] proposes a multi-task pointwise network for semantic-instance segmentation and a multi-value conditional random field model for the joint optimization. [24] learns global features from an intermediate bird's-eye view representation and then propagates the learned features to 3D point clouds. [25] processes a voxelized point cloud and predicts the instance affinity between neighboring voxels at different scales. [26] directly regresses 3D bounding boxes for all instances in a point cloud and simultaneously predicts a point-level mask for each instance.

B. Deep Learning on Point Clouds

Deep learning on point clouds develops fast in recent years. View-based methods [27], [28] convert point clouds to images and apply 2D convolution. The voxel-based method [29], [30] is a natural generalization of 2D convolution. However, the performance of the voxel-based methods is limited by the resolution of the voxels. [31], [32] exploit the sparsity property of 3D data and enable a 3DCNN to achieve higher resolution and efficiency. Additionally, such sparse convolutional operations can be easily combined with many useful network frameworks for 2D images. PointNet [33] provides a brand new direction for 3D deep learning. It directly processes raw point clouds without quantitative errors. Several methods [34]–[36] model the local relationship to extract hierarchical features using PointNet. The point-based methods can not process large scale scenes with too many points, so most of them divide the whole scene into a collection of blocks instead of directly processing the whole scene. Such strategy can be effective for tasks such as semantic segmentation. However, for instance segmentation, the division makes the post-processing pipeline much more complex since we need to merge the same instance from different blocks. As a result, the voxel-based method is chosen to be the backbone of our network to generate the instance embeddings.

III. METHOD DESCRIPTION

In section III-A, we first describe the whole network architecture. In section III-B, we introduce our proposed structure-aware loss function for supervised learning of instance embeddings. In section III-C, we present the attention-based kNN for automatically selecting information from neighbors and eliminating the quantization error.

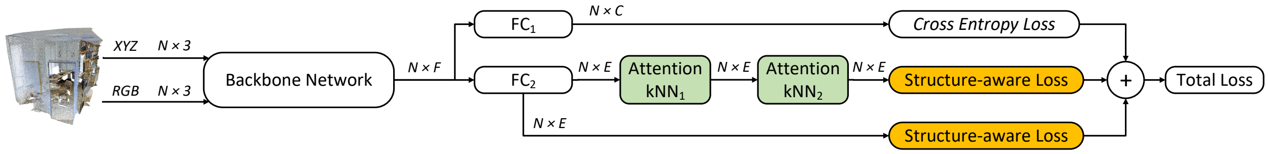

Fig. 2. Illustration of the whole network architecture. N is the number of points. F is the dimension of the backbone output. C is the number of the semantic classes. E is the dimension of the instance embedding. The mean-shift algorithm is used to cluster the instance embeddings during the inference.

A. Network Architecture

The whole network (illustrated in Fig. 2) consists of three main components including the backbone network, the structure-aware loss function and the attention-based kNN algorithm. The backbone network is a kind of 3DCNN and borrowed from [32]. The input is the original point coordinates with RGB attributes. We recommend [32] for more details on the backbone network. The output of the backbone network is fed into two branches. The semantic branch directly outputs the semantic predictions supervised by the cross entropy loss function after a fully connected layer. The instance branch outputs the initial instance embeddings for the structure-aware loss function after a fully connected layer. To eliminate the quantization error caused by 3D voxel, the output of the fully connected layer is fed into a series of attention-based kNN modules. Finally, three loss items are added as the total loss. During the inference, the mean-shift algorithm is used to cluster the embeddings generated by the instance branch to obtain instance clusters.

B. Structure-Aware Loss Function

After generating the initial embeddings for all points, we hope that the points within the same instance have similar embeddings while the points from different instances are apart in the embedding space. This is a classical problem in metric learning [13], [37]. However, for a 3D point cloud, it does not only have the embedding information but also have the 3D geometric relations. This is different from past research on metric learning. We utilize this relation to make the final results more discriminative.

On the one hand, we want to minimize the distance between embeddings within the same instance. The Euclidean distance is chosen to measure the similarity due to its simplicity. For the i_{th} instance, we calculate the geometric center $\mu_{p,i}$ and the embedding center $\mu_{s,i}$, and then the corresponding centers are subtracted from the 3D coordinates and the embedding vector, respectively:

$$\mu_{p,i} = \frac{1}{N_i}\sum_{k=1}^{N_i} p_{i,k}, \quad p_{i,k}^{'} = p_{i,k} - \mu_{p,i} \tag{1}$$

$$\mu_{s,i} = \frac{1}{N_i}\sum_{k=1}^{N_i} s_{i,k}, \quad s_{i,k}^{'} = s_{i,k} - \mu_{s,i} \tag{2}$$

where $p_{i,k}$ and $s_{i,k}$ are the 3D coordinates and the embedding of the k_{th} point within the i_{th} instance, respectively. $p_{i,k}^{'}$ and $s_{i,k}^{'}$ are the 3D coordinates and the embedding after mean centering, respectively.

The intra-loss item for the i_{th} instance is formalized as follows:

$$Loss_i^{intra} = \sum_{k=1}^{N_i} \mathrm{g}(\|p_{i,k}^{'}\|)[\|s_{i,k}^{'}\| - \alpha]_+^2 \tag{3}$$

where α is a threshold for penalizing large embedding distances. N_i is the point number of the i_{th} instance. $\mathrm{g}(x) : R \mapsto R$ is a function that is monotonically increasing. It means that points far from the geometric center are penalized more heavily in Equation (3). We use the sigmoid function as $\mathrm{g}(x)$ in our application. $[x]_+$ means $max(0, x)$.

It should be noted that the embeddings of the points near the edge are more likely to be different from the mean embedding which means that they are hard examples for instance segmentation. Equation (3) focuses more on these points. Experiments show the effectiveness of this operation.

On the other hand, to make the points of different instances discriminative, the mean embeddings between different instances should be far from each other:

$$Loss_{ij}^{inter} = [\beta - \|\mu_{s,i} - \mu_{s,j}\|]_+^2 \tag{4}$$

where β is a threshold for the distance between mean embeddings. It means that the loss function only penalizes small distances. If a distance is larger than the threshold, then it will not contribute to the loss value since the embeddings are far enough apart in the embedding space.

The final loss function is composed of the above items:

$$Loss = \frac{1}{M}\sum_{i=1}^{M} Loss_i^{intra} + \frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j=1, j\neq i}^{M} Loss_{ij}^{inter} \tag{5}$$

where M is the total number of instances in the scene.

Analysis. We define an attractive term and a repulsive term in Equation (5) in line with [9] and utilize the 3D information to enhance the discrimination of the instance embedding. The effectiveness of the 3D structure is shown in two aspects. First, we extend the attractive term in Equation (3) with a spatial weighting related to the geometric center. The usage of the spatial weighting makes the network focus more on the points near the edge which are hard examples in the training process. Second, the raw input of our network includes the 3D point coordinates which makes the embedding reflect the spatial relation. Objects are separate in 3D space, and the 3D coordinates reflect the structure of 3D objects. However, in 2D image segmentation, the input is the RGB value, which

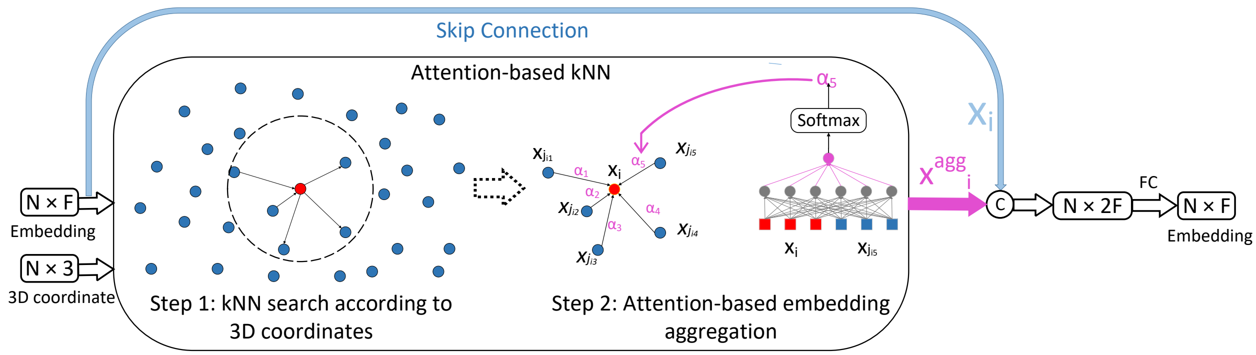

Fig. 3. Illustration of the attention-based kNN. In step 1, for each input point, the k-nearest neighbors are searched according to the 3D coordinates. In step 2, different weights are learned for different neighbors. The value of the weight is determined by both the center point and the corresponding neighbor. The output of the attention-based kNN is the weighted average of the embeddings of k neighbors. The skip connection is used to concatenate the output and the input embeddings together. A fully connected layer follows to update the output embedding.

can not directly reflect the spatial geometric information. We can find that proposal-free methods usually perform worse than proposal-based methods in 2D segmentation while our method can perform better than some proposal-based methods [19], [20] in 3D segmentation.

C. Attention-based K-Nearest Neighbor

3DCNN is a powerful method for extracting features from point clouds. However, 3DCNN is a voxel-based method. One voxel may contain several points with different categories. Regardless of how well the method extracts features, a quantization error always exists. The k-nearest neighbor (kNN) algorithm is a pointwise operation. A point can aggregate information from surrounding neighbors. Such a pointwise operation can eliminate the quantization error caused by 3D voxel. However, because the neighbors are searched according to the 3D coordinates, aggregating information by simple averaging will result in some wrong information if the neighbors do not belong to the same instance as the center point. Therefore, we propose an attention-based kNN for automatic embedding selection. We add the attention-based kNN after the 3DCNN to refine the instance embedding. Combining the 3DCNN and the attention-based kNN can allow them to benefit from each other. In this section, we describe the attention-based kNN module. Fig. 3 illustrates the process of the attention-based kNN.

The input embeddings of point clouds are denoted by $X = \{x_1, ..., x_n\} \subseteq R^F$. $\left\{x_{j_{i_1}}, ..., x_{j_{i_k}}\right\}$ are the k-nearest neighbors of x_i according to their 3D coordinates. The average-based kNN aggregation process can be formalized as follows:

$$x_i^{aggregate} = \frac{1}{k}\sum_{m=1}^{k} x_{j_{i_m}} \tag{6}$$

For automatic embedding selection and aggregation, we utilize the attention mechanism. The operation can be formalized as follows:

$$x_i^{aggregate} = \sum_{m=1}^{k} \alpha_m x_{j_{i_m}} \tag{7}$$

where α_m is the attention weight for each neighbor. It is related to the embedding of the neighbor and the corresponding center point and can be calculated as follows:

$$p_m = f(x_i, x_{j_{i_m}}) \tag{8}$$

where $f : R^{2\times F} \mapsto R^1$ is a two-layer fully connected network. α_m is the normalization of p_m using the softmax function:

$$\alpha_m = \mathrm{softmax}(p_m) = \frac{\exp(p_m)}{\sum_{m=1}^{k}\exp(p_m)} \tag{9}$$

After aggregating the embeddings from neighbors, the skip connection is used to combine the original embedding and the aggregated embedding together to obtain the hierarchical embedding. Then, a fully connected layer is used to generate the final output embedding:

$$x_i^{output} = [x_i, x_i^{aggregate}]W \tag{10}$$

where $W \subseteq R^{2F\times F}$ is a trainable parameter.

Analysis. Compared to the average-based kNN aggregation, the attention-based kNN can learn different weights for different neighbors. This means that the relation between two points is determined by their features. Additionally, the attention-based kNN is a point-based operation while the backbone is a voxel-based network. Combining the two representations can allow them to benefit from each other. The voxel-based method has the ability to effectively process large scenes and the point-based method can eliminate the quantization error caused by voxel and provide a more precise receptive field since it directly uses the absolute coordinates to search neighbors.

Implementation. We search the k-nearest neighbors using CUDA implementation on the GPU. For each point, the features of its k neighbors are obtained. Equation (7) aggregates these features in an attention fashion. The kNN search itself does not have learnable parameters, so it does not need to be updated when back propagating. The role of the kNN search is to provide the index of neighbors. Our attention-based kNN can be trained in an end-to-end style without any special operation. The function $f : R^{2\times F} \mapsto R^1$ in Equation (8) needs to be updated when back propagating. Its training process has no difference from other networks.

IV. EXPERIMENTS

Datasets. We evaluate our model with two datasets which provide 3D instance segmentation labels:

- ScanNet [11]: This dataset contains 1613 3D indoor scans. We follow the official split of 1201 training samples, 300 validation samples and 100 testing samples (without ground truths). The dataset provides a benchmark for several tasks including 3D instance segmentation. It provides images from different views but we only use the point cloud data in our method.
- NYUv2 [14]: This dataset contains 1449 single RGB-D images. We follow the same preprocessing method as [12] and [19] to obtain the 3D annotations of point clouds. We follow the standard split of 795 training samples and 654 testing samples.

Implementation Details. We implement the network with Pytorch1.0 [38] and run it on a single NVIDIA GTX1080Ti. Our network can be easily trained in an end-to-end style. We use the ADAM optimizer with a learning rate of 0.001. α and β in the structure-aware loss function are set to 0.7 and 1.5 respectively. For the kNN search, we set $k = 8$ and use the L_2-distance according to the 3D coordinates of points. The dimension of the instance embedding is set to 4. We find that higher dimension has minor influence on the accuracy and makes it difficult to determine the bandwidth of the mean-shift algorithm during the inference. The mean-shift algorithm is used to cluster the embeddings generated by the instance branch to obtain instance clusters. The bandwidth of the mean-shift algorithm is set as 1.0. Meanwhile, the semantic branch outputs a semantic prediction for each point. We determine the category of each instance cluster by majority vote. For data augmentation, we randomly scale the scenes and rotate them along the vertical axis.

In our experiment, we use two backbone networks with different model capacities provided by [32]. The first backbone network is a UNet-like architecture based on the submanifold sparse convolution with a smaller capacity and a faster speed. The second is a ResNet-like architecture with a larger capacity and a slower speed. In addition to UNet with the standard sparse convolutional layers, the ResNet-like architecture uses ResNet style convolutional layers[1] for deeper feature extraction. We train the whole model from scratch with the UNet backbone for 650 epochs and the model with the ResNet backbone for 300 epochs.

Metrics. The average precision (AP) is widely used in instance segmentation. For the ScanNet online benchmark, it provides the result of the AP with an IoU threshold of 0.5 ($AP_{0.5}$) and 0.25 ($AP_{0.25}$) and the mean AP (mAP) with the IoU ranging from 0.5 to 0.95 [11]. For the NYUv2 dataset, the AP with an IoU threshold of 0.25 ($AP_{0.25}$) is commonly used. For both datasets, images and RGB-D point clouds are provided. Some previous methods use both inputs while others use a single input. Our network only uses the point clouds as the input. We do not use the features extracted from images via image-based 2D networks.

A. Instance Segmentation on ScanNet

The ScanNet dataset provides an online benchmark and we first evaluate our method with it. Eighteen categories are used in the instance segmentation task which makes it more challenging compared to instance segmentation on a single category.

Among all previous methods, SGPN [12] is the most similar method to our method. Compared to SGPN, the space complexity and the computational complexity of our proposed structure-aware function are both $\mathcal{O}(n)$, while those of SGPN are both $\mathcal{O}(n^2)$. Additionally, our proposed function considers the structure information while SGPN considers each point equivalently. R-PointNet [19], 3D-SIS [20], PanopticFusion [22] and 3D-BoNet [26] are proposal-based methods. 3D-SIS and PanopticFusion use not only the point cloud but also images from multiple views as the input. The image information also contributes to their final results.

Our method achieves state-of-the-art performance on the ScanNet benchmark. Tab. I provides the $AP0.5$ of each class and the overall $AP0.5$. Tab. II provides the overall mAP, $AP0.5$ and $AP0.25$ reported on the ScanNet online benchmark. The qualitative results are shown in Fig 4. The model using UNet as the backbone outperforms most of the methods including R-PointNet [19]. The model using ResNet as the backbone almost outperforms all methods except for the $AP0.5$ of PanopticFusion-inst [22] which additionally uses images as the input and 3D-BoNet [26]. However, our method performs better than them on the mAP and $AP0.25$ metrics. Specially, our method exceeds PanopticFusion-inst on mAP by a large margin. Compared to the $AP0.5$, the mAP requires a higher overlap between the prediction and the ground truth. This means that the prediction of our method is more complete than that of PanopticFusion-inst. This is attributed to the framework of our method. The proposed structure-aware loss function aims to make the points within the same instance have similar embeddings. Moreover, it focuses more on hard examples far from the geometric center. Therefore, our method is more likely to generate similar embeddings for all points within an object. Additionally, the attention-based kNN further enhances the local consistency of the instance embeddings.

In Tab. I, we find that classes such as pictures, counters, bookshelves, and desks are not predicted well. For 3D points, pictures and bookshelves are easily misclassified as the wall because their geometric shapes are similar. Desks are also easily predicted as tables. In contrast, panopticFusion-inst well predicts windows, bookshelves and pictures. These classes are easily recognized in images.

[1]https://github.com/facebookresearch/SparseConvNet/blob/master/examples/ScanNet/README.md

TABLE I

RESULTS ON THE TEST SET OF THE SCANNET (V2) 3D INSTANCE SEGMENTATION BENCHMARK. $AP_{0.5}$ IS REPORTED IN THE TABLE.

Method	image	point cloud	Mean	cabi-net	bed	chair	sofa	table	door	win-dow	book-shelf	pic-ture	coun-ter	desk	cur-tain	fri-dge	show-er	toilet	sink	bath-tub	other
Mask R-CNN [7]	yes	no	5.8	5.3	0.2	0.2	10.7	2.0	4.5	0.6	0.0	23.8	0.2	0.0	2.1	6.5	0.0	2.0	1.4	33.3	2.4
SGPN [12]	no	yes	14.3	6.5	39.0	27.5	35.1	16.8	8.7	13.8	16.9	1.4	2.9	0.0	6.9	2.7	0.0	43.8	11.2	20.8	4.3
3D-BEVIS [24]	no	yes	24.8	3.5	56.6	39.4	60.4	18.1	9.9	17.1	7.6	2.5	2.7	9.8	3.5	9.8	37.5	85.4	12.6	66.7	3.0
R-PointNet [19]	no	yes	30.6	34.8	40.5	58.9	39.6	27.5	28.3	24.5	31.1	2.8	5.4	12.6	6.8	21.9	21.4	82.1	33.1	50.0	29.0
3D-SIS [20]	yes	yes	38.2	19.0	43.2	57.7	**69.9**	27.1	32.0	23.5	24.5	7.5	1.3	3.3	26.3	42.2	**85.7**	88.3	11.7	**100.0**	24.0
MASC [25]	no	yes	44.7	**38.2**	55.5	**63.3**	63.9	38.6	36.1	27.6	38.1	32.7	0.2	26.0	50.9	45.1	57.1	98.0	36.7	52.8	43.2
PanopticFusion [22]	yes	yes	47.8	25.9	71.2	55.0	59.1	26.7	25.0	35.9	**59.5**	**43.7**	0.0	17.5	61.3	41.1	**85.7**	**94.4**	**48.5**	66.7	43.4
3D-BoNet [26]	no	yes	**48.8**	30.1	67.2	48.4	49.9	51.3	34.1	**43.9**	59.0	12.5	9.8	**30.6**	**62.0**	**43.4**	79.6	90.9	40.2	**100.0**	25.9
UNet+strucLoss+kNN	no	yes	31.9	18.9	71.5	47.9	61.5	35.5	20.1	9.3	23.3	10.7	0.8	6.7	21.8	12.3	43.8	91.6	15.0	66.7	17.3
ResNet+strucLoss+kNN	no	yes	45.9	25.9	**73.7**	58.7	53.6	**59.0**	**41.6**	30.4	15.9	12.8	**13.8**	21.7	47.5	31.5	71.4	87.3	41.1	**100.0**	**40.8**

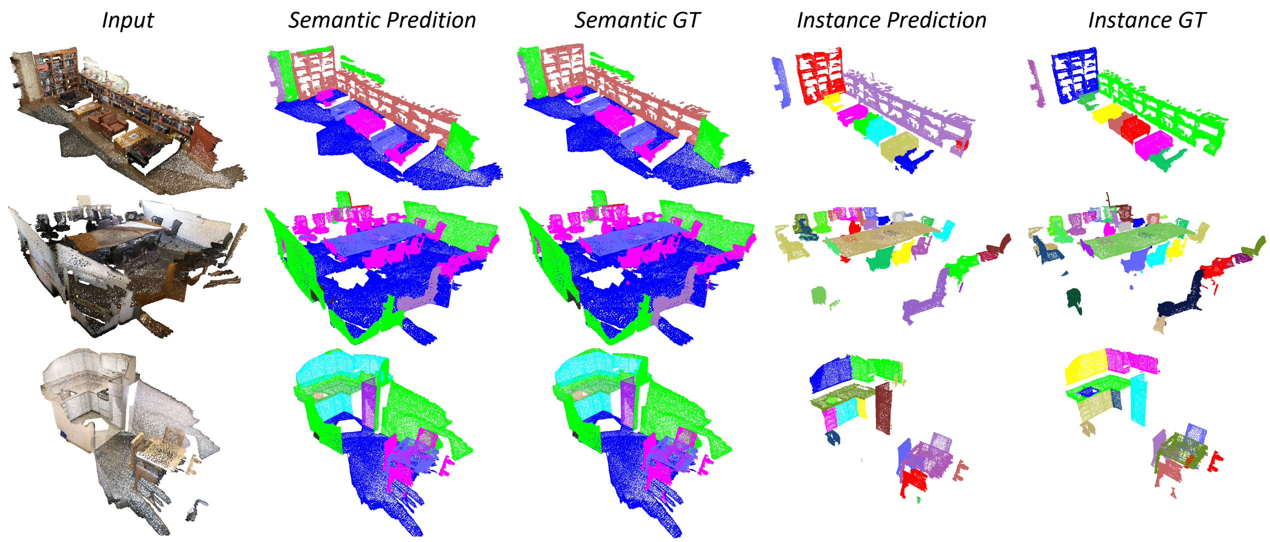

Fig. 4. Visualization of ScanNet results. The first column is the input of our model. The second column is the prediction of the semantic labels. The third column is the ground truth of the semantic segmentation. The fourth column is the instance prediction. The fifth column is the ground truth of the instance segmentation. For the instance segmentation, we only visualize the 18 categories that are useful for the evaluation while dropping the other categories.

TABLE II

RESULTS ON THE TEST SET OF THE SCANNET (V2) 3D INSTANCE SEGMENTATION BENCHMARK. mAP, $AP_{0.5}$ AND $AP_{0.25}$ ARE REPORTED IN THE TABLE.

Method	mAP	AP0.5	AP0.25
Mask R-CNN [7]	2.2	5.8	26.1
SGPN [12]	4.9	14.3	39.0
3D-BEVIS [24]	11.7	24.8	40.1
R-PointNet [19]	15.8	30.6	54.4
3D-SIS [20]	16.1	38.2	55.8
MASC [25]	25.4	44.7	61.5
PanopticFusion [22]	21.4	47.8	69.3
3D-BoNet [26]	25.3	**48.8**	68.7
UNet+strucLoss+kNN(ours)	16.1	31.9	60.5
ResNet+strucLoss+kNN(ours)	**26.3**	45.9	**69.5**

B. Instance Segmentation on NYUv2

Different from the ScanNet dataset, the NYUv2 dataset provides single RGB-D images instead of whole scenes. Previous methods usually use both images and point clouds as the input to increase the precision on this dataset. We only use the 3D point clouds as the input in this paper. Despite this, our method outperforms all state-of-the-art methods on this dataset (shown in Tab. III). Especially, our method achieves the highest precision for many categories. Since the NYUv2 dataset provides single RGB-D images with partial point clouds, categories such as boxes, monitors, and garbage bins are difficult to recognize only using point clouds. It is easier to segment these categories on the image than on the point cloud. Therefore, MRCNN obtains better results than our method on some of these categories. Fusing visual features from images can also be helpful. We leave the multi-sensor fusion as a future work.

C. Ablation Study

We conduct an ablation study on the validation set of the ScanNet (v2) dataset.

Different loss functions. To validate the effectiveness

TABLE III

RESULTS ON THE TEST SET OF THE NYUV2 DATASET. $AP_{0.25}$ IS REPORTED IN THE TABLE.

Method	image	point cloud	mean	bath-tub	bed	book-shelf	box	chair	coun-ter	desk	door	dres-ser	gar-bage	lamp	moni-tor	night stand	pil-low	sink	sofa	table	TV	toilet
MRCNN	yes	no	29.3	26.3	54.1	23.4	3.1	39.3	34.0	6.2	17.8	23.7	**23.1**	31.1	35.1	25.4	26.6	36.4	47.1	21.0	**23.3**	58.8
MRCNN*	yes	no	31.5	24.7	66.3	20.1	1.4	44.9	43.9	6.8	16.6	29.5	22.1	29.2	**29.3**	36.9	34.6	37.1	48.4	26.6	21.9	58.5
SGPN-CNN [12]	yes	yes	33.6	45.3	62.5	43.9	0.0	45.6	40.7	30.0	20.2	42.6	8.8	28.2	15.5	43.0	30.4	51.4	**58.9**	25.6	6.6	39.0
R-PointNet-CNN [19]	yes	yes	39.3	62.8	51.4	35.1	**11.4**	54.6	45.8	**38.0**	22.9	**43.3**	8.4	36.8	18.3	**58.1**	**42.0**	45.4	54.8	**29.1**	20.8	67.5
ResNet+strucLoss+kNN	no	yes	**43.0**	**82.1**	**67.3**	**48.1**	3.5	**65.4**	**56.8**	14.5	**37.6**	23.1	7.3	**60.0**	4.4	52.9	34.3	**68.2**	55.0	28.3	20.7	**87.2**

TABLE IV

COMPARISON OF DIFFERENT LOSS FUNCTIONS.

Method	AP	AP0.5	AP0.25
UNet+intra-only	7.5	18.4	41.6
UNet+inter-only	6.3	16.5	42.0
UNet+vanillaLoss	15.0	33.8	59.9
UNet+strucLoss	**15.8**	**35.0**	**61.3**

TABLE V

COMPARISON OF DIFFERENT KNN LAYERS. AVERAGE-KNN ASSIGNS EQUAL WEIGHTS FOR EACH NEIGHBOR. ATTENTION-KNN ASSIGNS LEARNABLE WEIGHTS FOR EACH NEIGHBOR.

Method	AP	AP0.5	AP0.25
UNet+strucLoss	15.8	35.0	61.3
UNet+strucLoss+average-kNN×1	16.0	34.8	62.0
UNet+strucLoss+average-kNN×2	15.1	34.5	61.7
UNet+strucLoss+attention-kNN×1	16.3	35.6	62.1
UNet+strucLoss+attention-kNN×2	**17.1**	**36.0**	**63.0**
UNet+strucLoss+attention-kNN×3	16.5	35.1	62.3

TABLE VI

COMPARISON OF DIFFERENT METHODS FOR FEATURE AGGREGATION.

Method	AP	AP0.5	AP0.25
UNet+strucLoss+mlp-pooling×2	16.6	35.4	62.3
UNet+strucLoss+attention-kNN×2	**17.1**	**36.0**	**63.0**

TABLE VII

COMPARISON OF DIFFERENT BACKBONES.

Method	AP	AP0.5	AP0.25
UNet+strucLoss+attention-kNN×2	17.1	36.0	63.0
ResNet+strucLoss+attention-kNN×2	**27.0**	**46.4**	**67.2**

of our structure-aware loss function, we compare it with the vanilla loss function. The vanilla loss function can be viewed as a center loss [13] with an additional inter-item. The vanilla version does not use structure information, so the importance of each point in the same instance is the same. Tab. IV shows the comparison. The use of the structure-aware loss function increases the $AP_{0.5}$ and $AP_{0.25}$ by more than 1%. This means that assigning larger weights to points far from the geometric center is beneficial to generating more discriminative embeddings. Additionally, we train the network on the intra-cluster and inter-cluster loss separately. We find that using the single constraint can not obtain satisfactory results. If using the intra-cluster loss solely, the points from different instances may have similar embeddings. If using the inter-cluster loss solely, the embeddings of points within the same instance may be of great difference. Therefore, the two items need to be applied simultaneously.

Different kNN layers. We compare different types and numbers of kNN layers to explore their effectiveness. Tab. V provides the results. The average kNN assigns equal weights for each neighbor. The results show that it barely improves the result or even harms it. We suppose that simple averaging may cause blurry features. However, the attention-based kNN can contribute to the final result due to its automatic embedding selection mechanism. It can be found that the use of two layers performs the best. More layers do not necessarily lead to better results. We find that using more layers may cause oversmoothing. Additionally, too many layers may increase the difficulty of training. Further, we compare the proposed attention-based kNN with the feature aggregation method in PointNet++. The feature aggregation operation in PointNet++ uses MLP to update features and max pooling to aggregate features. We call the operation mlp-pooling. Here our goal is to refine the features. We think that weighted averaging has better interpretability than pooling. Tab. VI also proves the effectiveness of our method.

Different backbone networks. Our proposed architecture can adapt to different backbone networks. In this paper, we compare two models that use the UNet backbone and the ResNet backbone. The results are shown in Tab. VII. The model using the ResNet backbone outperforms the model using the UNet backbone. This means that deeper models can perform better using our framework which shows the expansibility of our method.

V. CONCLUSIONS

In this paper, we propose a framework for 3D point cloud instance segmentation. By using the proposed structure-aware loss function, discriminative instance embeddings can be easily generated. To aggregate information from neighbors and eliminate the quantization error caused by 3D voxel, the attention-based kNN is proposed to learn different weights for different neighbors. Experiments show that our approach achieves state-of-the-art performance on the ScanNet benchmark and the NYUv2 dataset. In the future, multi-sensor fusion can be added into our network to combine geometric features and image features.

References

[1] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[8] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy, "Semantic instance segmentation via deep metric learning," *arXiv preprint arXiv:1703.10277*, 2017.

[9] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017.

[10] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9018–9028.

[11] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[12] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2569–2578.

[13] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[14] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[19] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "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, pp. 3947–3956.

[20] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3d semantic instance segmentation of rgb-d scans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4421–4430.

[21] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "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, pp. 909–918.

[22] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[23] Q.-H. Pham, T. Nguyen, B.-S. Hua, G. Roig, and S.-K. Yeung, "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, pp. 8827–8836.

[24] C. Elich, F. Engelmann, J. Schult, T. Kontogianni, and B. Leibe, "3D-BEVIS: Birds-Eye-View Instance Segmentation," in *German Conference on Pattern Recognition*, 2019, pp. 48–61.

[25] C. Liu and Y. Furukawa, "Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation," *arXiv preprint arXiv:1902.04478*, 2019.

[26] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3d instance segmentation on point clouds," in *Advances in Neural Information Processing Systems*, 2019, pp. 6737–6746.

[27] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[28] A. Boulch, B. Le Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks." in *3DOR*, 2017.

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

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

[31] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3, 2017.

[32] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[35] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in neural information processing systems*, 2018, pp. 820–830.

[36] Z. Liang, M. Yang, L. Deng, C. Wang, and B. Wang, "Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8152–8158.

[37] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.