

JSNet: Joint Instance and Semantic Segmentation of 3D Point Clouds

Lin Zhao, Wenbing Tao*

National Key Laboratory of Science and Technology on Multispectral Information Processing
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China
{linzhao, wenbingtao}@hust.edu.cn

Abstract

In this paper, we propose a novel joint instance and semantic segmentation approach, which is called JSNet, in order to address the instance and semantic segmentation of 3D point clouds simultaneously. Firstly, we build an effective backbone network to extract robust features from the raw point clouds. Secondly, to obtain more discriminative features, a point cloud feature fusion module is proposed to fuse the different layer features of the backbone network. Furthermore, a joint instance semantic segmentation module is developed to transform semantic features into instance embedding space, and then the transformed features are further fused with instance features to facilitate instance segmentation. Meanwhile, this module also aggregates instance features into semantic feature space to promote semantic segmentation. Finally, the instance predictions are generated by applying a simple mean-shift clustering on instance embeddings. As a result, we evaluate the proposed JSNet on a large-scale 3D indoor point cloud dataset S3DIS and a part dataset ShapeNet, and compare it with existing approaches. Experimental results demonstrate our approach outperforms the state-of-the-art method in 3D instance segmentation with a significant improvement in 3D semantic prediction and our method is also beneficial for part segmentation. The source code for this work is available at <https://github.com/dlinzhao/JSNet>.

Introduction

Semantic segmentation is the task which is used to segment all informative regions in a scene and classify each region into a specific class. Instance segmentation is different from semantic segmentation for that different objects of the same class will have different labels. Both the two tasks have a wide applications in real-world scenarios, e.g., autonomous driving and mobile-based navigation. In 2D images, those two tasks have achieved remarkable results (Chen et al. 2018; He et al. 2017; Li et al. 2019). However, the studies of 3D semantic and instance segmentation are still facing huge challenges, e.g., large-scale with noisy data processing, high computation as well as memory consumption.

*Corresponding author.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

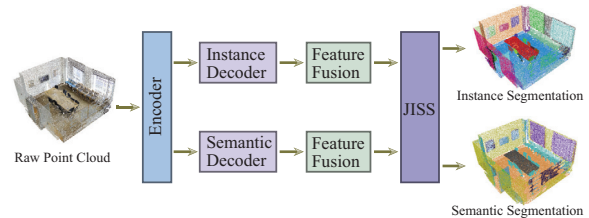


Figure 1: Our network JSNet takes raw point clouds as inputs and gets outputs instance and semantic segmentation results for each point. JISS stands for Joint Instance and Semantic Segmentation.

Literature research shows that 3D scene data have different representations, e.g., volumetric grids (Wu et al. 2015; Thanh Nguyen et al. 2016; Maturana and Scherer 2015) and 3D point clouds (Qi et al. 2017b; Li et al. 2018; Wang et al. 2018b; Yu et al. 2018). Compared with other representations, point cloud is a more compact and intuitive representation of 3D scene data. Recently, more efficient and powerful deep learning network architectures (Qi et al. 2017b; Wu, Qi, and Fuxin 2019; Li, Chen, and Hee Lee 2018) have been proposed to directly process point clouds and shown promising results in point cloud classification and part segmentation. Those approaches are often used as feature extraction network in other tasks, e.g., instance segmentation and semantic segmentation.

In previous works, instance segmentation and semantic segmentation have often been processed respectively or instance segmentation is treated as a post-processing task of semantic segmentation (Wang et al. 2018a; Pham et al. 2019a). However, those two problems are related because points of different categories belong to different instances and points of the same instance belong to the same class. Recently, (Pham et al. 2019b) handles the two problems with multi-task pointwise network and multi-value Conditional Random Field (CRF). However, the CRF is an individual part behind the Convolutional Neural Network (CNN), it is difficult to explore the performance of their combination. Moreover, this method does not investigate whether semantic segmentation and instance segmentation can facili-

tate each other. At the same time, ASIS (Wang et al. 2019b) is proposed to address the two tasks simultaneously, which adapts the semantic feature to instance feature space by a fully connected layer and aggregates instance feature to semantic feature space by K Nearest Neighbor (kNN). However, the performance of this method is limited because it is difficult to choose the right K value and distance metric for kNN. Besides, it has expensive computation and memory consumption because it will generate a high order sparse tensor during training process.

In this work, we introduce a joint instance semantic segmentation neural network of 3D point clouds called JSNet to address the two fundamental problems: semantic segmentation and instance segmentation. The proposed network JSNet includes four parts: a shared feature encoder, two parallel branch decoders, a feature fusion module for each decoder, a joint segmentation module. The feature encoder and decoders are built based on PointNet++ (Qi et al. 2017b) and PointConv (Wu, Qi, and Fuxin 2019) to learn more effective high-level semantic features. To obtain more discriminative features, we propose a point cloud feature fusion module to fuse the high-level and low-level information to refine the output features. In order to make the two tasks promote each other, a novel joint instance and semantic segmentation module is proposed to handle instance and semantic segmentation simultaneously. Specifically, this module transforms semantic features into instance embedding space by a 1D convolution and then the transformed features are further fused with instance features to facilitate instance segmentation. Meanwhile, this module also aggregates instance features into semantic feature space by implicit learning to promote semantic segmentation. Thus, our approach can be used to learn instance-aware semantic fusion features and semantic-aware instance embedding features, which can make the predictions of those points more accurate.

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

- We design a more efficient Point Cloud Feature Fusion (PCFF) module to generate more discriminative features and improve the accuracy of point predictions.
- We propose a novel Joint Instance and Semantic Segmentation (JISS) module to make instance segmentation and semantic segmentation mutual promote. This module further improve the accuracy with acceptable GPU memory consumption during training process.
- We achieve impressive results on the S3DIS dataset (Armeni et al. 2016) along with a significant improvement on the 3D instance segmentation task. Additionally, our experiments on the ShapeNet dataset (Yi et al. 2016) indicate that JSNet also can achieve satisfactory performance for part segmentation task.

Related Work

In this section, we briefly review some point cloud feature extraction works, and some existing approaches for semantic and instance segmentation in 3D scene. Especially, we concentrate on deep neural network-based methods applied

to 3D point clouds because of their proven robustness and efficiency in the field.

Deep learning for 3D Point Clouds

Although deep learning has been successfully used for 2D images, there are still many challenges in the feature learning capabilities of 3D point clouds with irregular data structures. Recently, PointNet (Qi et al. 2017a) is one of the first approaches of directly applying neural networks to point clouds. It uses shared Multi-Layer Perceptron (MLP) and max pooling to learn deep features from unordered point sets. However, PointNet has difficulty in capturing local region features. This drawback has been addressed by PointNet++ (Qi et al. 2017b) with a hierarchical neural network. The max pooling operation is a key structure to extract features from points for both PointNet and PointNet++. But it only keeps the strongest activation on a local or global region of feature maps, which may cause some useful detailed information lost for semantic and instance segmentation tasks.

Some later works (Simonovsky and Komodakis 2017; Hermosilla et al. 2018; Xu et al. 2018) extract features of point clouds with learning continuous filters for convolution calculation. The work (Simonovsky and Komodakis 2017) firstly presents the idea that learning continuous filters with edge-conditioned into 3D graph. Furthermore, Dynamic graph CNN (Wang et al. 2018b) introduces a method to update the graph dynamically. The following work PointConv (Wu, Qi, and Fuxin 2019) proposes an inverse density scale to re-weight the continuous function learned by MLP and compensate the non-uniform sampling, while it also needs high GPU memory during training process.

Semantic&Instance Segmentation on Point Clouds

For semantic segmentation, methods (Zhao et al. 2017; Chen et al. 2018) based on full convolutional networks (Long, Shelhamer, and Darrell 2015) have achieved tremendous progress in 2D domain. As for 3D semantic segmentation, 3D-FCNN introduced by (Huang and You 2016) predicts a coarse voxel label with a 3D fully convolutional neural network. SEGCloud (Tchapmi et al. 2017) extends 3D-FCNN with trilinear interpolation and fully connected conditional random fields. RNet (Huang, Wang, and Neumann 2018) models local dependencies for point clouds with a slice pooling layer, Recurrent Neural Network (RNN) layers, and a slice unpooling layer. 3P-RNN (Ye et al. 2018) models the inherent contextual features for semantic segmentation by using a pointwise pyramid pooling module and explores long-range spatial dependencies with two-direction hierarchical RNNs. Recently, GAC (Wang et al. 2019a), a graph attention convolution, is proposed to capture the structured feature of point clouds with dynamical kernels to adapt the structure of an object. However, there are few previous works which focus on semantics segmentation by using the advantages of instance embedding.

For instance segmentation, approaches (Li et al. 2019; Huang et al.) based on Mask R-CNN (He et al. 2017) dominate it on 2D images. However, there are few studies for 3D instance segmentation. SGPN (Wang et al. 2018a) generates instance proposals from learning a similarity matrix

of the point features with a double-hinge loss. GSPN (Yi et al. 2019) generates proposals by reconstructing shapes and outputs the final segmentation results based on PointNet++. 3D-BoNet (Yang et al. 2019) directly regresses 3D bounding boxes and predicts point-level masks for all instances simultaneously. Similarly, there are few works which segment instances using the advantages of semantic fusion.

However, most of the previous works tackle the two tasks separately. Very recently, (Pham et al. 2019b) proposes a multi-task pointwise network (MT-PNet) for predicting the semantic categories and instance embedding vectors and then uses a multi-value conditional random field (MV-CRF) as a post-processing. However, the CRF is an individual part behind the CNN, and it is difficult to explore the performance of their combination. Moreover, this method does not investigate whether semantic segmentation and instance segmentation can promote each other. Therefore, the performance improvement is not obvious. Meanwhile, ASIS (Wang et al. 2019b) is proposed to segment instances and semantics for 3D point clouds at once, which uses PointNet or PointNet++ as backbone network and then concatenates a proposed module ASIS. The ASIS adapts the semantic features to instance feature space by a fully connected layer and aggregates instance features to semantic feature space by kNN. While the approach (Wang et al. 2019b) has difficulty to choose the right K value and distance metric for kNN, and it also has high memory cost because it will produce a high order sparse matrix at training process.

Proposed Method

In this section, firstly, we describe the whole network architecture of our proposed JSNet for instance and semantic segmentation of 3D point clouds. Then, we elaborate on the two main components of our proposed network, including the Point Cloud Feature Fusion (PCFF) module and the Joint Instance and Semantic Segmentation (JISS) module, respectively.

Network Architecture

The whole network illustrated in Figure 2(a) composes with four main components including a shared encoder, two parallel decoders, a point cloud feature fusion module for each decoder, a joint segmentation module as the last part. For the two parallel branches, one aims to extract semantic feature for each point, while the other one is for instance segmentation task. Specifically for the feature encoder and two decoders, we can directly use PointNet++ or PointConv as our backbone network by duplicating a decoder because the two decoders have the same structure. However, as is mentioned above, as for instance or semantic segmentation, the PointNet++ may lose detailed information because of max pooling operation and the PointConv has expensive GPU memory consumption during training process. In this work, we combine the PointNet++ and PointConv to build a more effective backbone network with acceptable memory cost. The encoder of the backbone is built by concatenating a set abstraction module of PointNet++ and three feature encoding layers of PointConv. Similarly, the decoders are composed

with three depthwise feature decoding layers of PointConv followed by a feature propagation module of PointNet++.

For the whole pipeline, our network takes a point cloud of size N_a as input, then encodes it into a $N_e \times 512$ shaped matrix by the shared feature encoder. Next, the output of feature encoder is input into the two parallel decoders and processed by their following components separately. The semantic branch decodes the shared features and fuses the features of different layers into a semantic feature matrix F_{SS} shaped with $N_a \times 128$. Similarly, the instance branch outputs an instance feature matrix F_{IS} after the PCFF module. Finally, both the semantic features and the instance features are fetched and processed by the JISS module, and then output two feature matrices. One of the matrices P_{SSI} shaped with $N_a \times C$ which is used to predict the semantic categories, where C is the number of semantic categories. The other one E_{ISS} shaped with $N_a \times K$ is an instance feature matrix and it is used to predict the instance labels for each point, where K is the dimension of the embedding vector. In the embedding space, the embeddings represent the instance relationship of points: the points belonging to the same instance object are close, and the points of the different instances are kept away from each other.

At training time, the loss function \mathcal{L} of our network consists of semantic segmentation loss \mathcal{L}_{sem} and instance embedding loss \mathcal{L}_{ins} :

$$\mathcal{L} = \mathcal{L}_{sem} + \mathcal{L}_{ins}, \quad (1)$$

where \mathcal{L}_{sem} is defined with the classical cross entropy loss. As for the instance embedding loss, we utilize a discriminative function to express the embedding loss \mathcal{L}_{ins} inspired by the work in (De Brabandere, Neven, and Van Gool 2017). Specifically, the instance embedding loss function is formulated as follows:

$$\mathcal{L}_{ins} = \mathcal{L}_{pull} + \mathcal{L}_{push}, \quad (2)$$

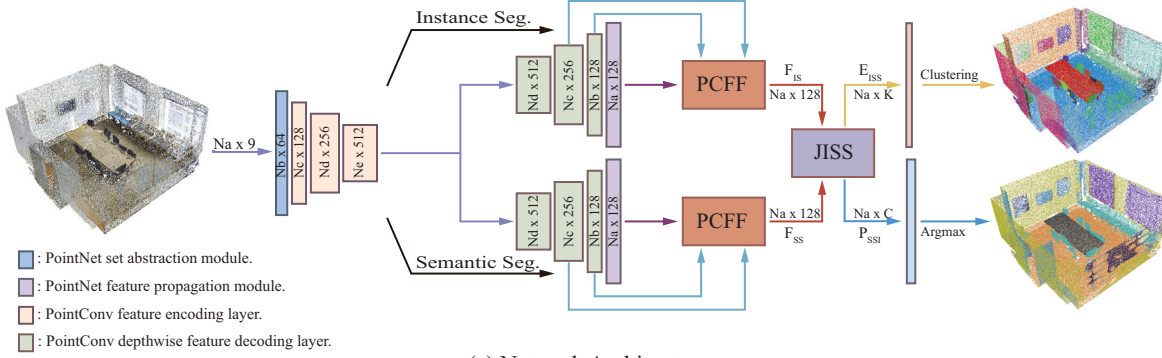
where \mathcal{L}_{pull} pulls embeddings close to the mean embedding of instance, while the \mathcal{L}_{push} makes the mean embedding of different instances separated from each other. Given the number of instances M , the number of elements N_m in the m -th instance, the embedding e_n of point, and the mean of embeddings μ_m in the m -th instance. Each term is rewritten as follows:

$$\mathcal{L}_{pull} = \frac{1}{M} \sum_{m=1}^M \frac{1}{N_m} \sum_{n=1}^{N_m} [\|\mu_m - e_n\|_1 - \delta_v]_+^2, \quad (3)$$

$$\mathcal{L}_{push} = \frac{1}{M(M-1)} \sum_{i=1}^M \sum_{\substack{j=1 \\ i \neq j}}^M [2\delta_d - \|\mu_i - \mu_j\|_1]_+^2, \quad (4)$$

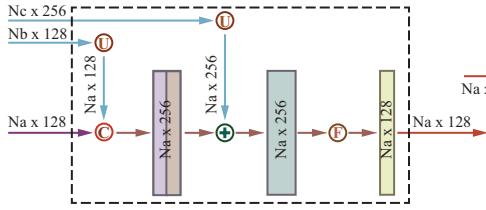
where $[x]_+ = \max(0, x)$; $\|\cdot\|_1$ is L_1 distance; δ_v and δ_d are margins for \mathcal{L}_{pull} and \mathcal{L}_{push} respectively.

At testing time, the final instance labels are generated by using a simple mean-shift clustering (Comaniciu and Meer 2002) on the embeddings and the final semantic categories are obtained by using a argmax operation.

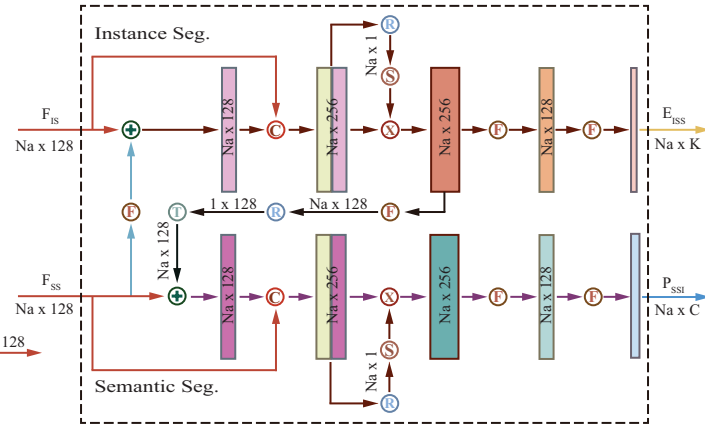


(a) Network Architecture

- Ⓚ: PointNet set abstraction module.
- Ⓛ: PointNet feature propagation module.
- Ⓜ: PointConv feature encoding layer.
- Ⓨ: PointConv depthwise feature decoding layer.
- Ⓤ: Upsampling tensor with interpolation.
- Ⓦ: Concatenates tensors along one dimension.
- Ⓧ: Element-wise addition.
- Ⓩ: Element-wise product.
- Ⓨ: ID Convolution with non-linear operation.
- Ⓛ: Mean of elements across dimension of a tensor.
- Ⓢ: Element-wise sigmoid.
- Ⓣ: Constructs a tensor by tiling a given tensor.



(b) Point Cloud Feature Fusion Module



(c) Joint Instance and Semantic Segmentation Module

Figure 2: An overview of the Joint Instance Semantic Segmentation Neural Network of 3D Point Cloud (JSNet). (a) Illustration of the network architecture. (b) Components of the Point Cloud Feature Fusion (PCFF) module. (c) Components of the Joint Instance and Semantic Segmentation (JISS) module. Different colored blocks represent different modules in (a), while those blocks represent different features in (b) and (c).

Point Cloud Feature Fusion Module

In the segmentation and detection tasks for 2D image, only the feature of last layer is used for prediction in previous works, while different layer features are fused in subsequent approaches (Lin et al. 2017; He et al. 2017; Chen et al. 2018) because the high level layer has richer semantic information while the low level has much more detailed information. Those works indicate that the fused features are beneficial for better prediction.

Based on observation above, we propose a Point Cloud Feature Fusion (PCFF) module for semantic and instance segmentation in point clouds. Figure 2(b) presents the details of the structure. Considering the precision, computation and GPU memory consumption, we only fuse the last three layers of the decoder. We use F_a , F_b and F_c to represent those feature matrices of the decoder with shape $N_a \times 128$, $N_b \times 128$ and $N_c \times 256$ respectively. Firstly, we concatenate F_a and F_b' upsampling with interpolation from F_b . Then the former output is added to F_c' (upsampling from F_c) element-wise and a convolution is applied to the previous result. Following (Qi et al. 2017b), the interpolation is achieved by using an inverse square distance weighted average based on

three nearest neighbors. Finally, the PCFF generates a fused feature matrix with shaped $N_a \times 128$. This module can refine the output features from the decoder with acceptable computation and memory consumption.

Joint Instance and Semantic Segmentation

In fact, both the semantic segmentation and the instance segmentation map the initial point cloud features to different new high-level feature spaces separately. In the semantic feature space, points of the same semantic category are clustered together, while the different classes are separated. In the instance feature space, points of the same instance object are closely assembled, while points of different instances are separated. It indicates that we could extract semantic awareness information from the semantic feature space to integrate the information into the instance features and generates semantic-aware instance embedding features, and vice versa.

Based on this observation, we propose a Joint Instance Semantic Segmentation (JISS) module to obtain semantic labels and segment instance objects simultaneously, as is illustrated in Figure 2(c). The JISS module transforms se-

semantic features into instance embedding space and then the transformed features are further fused with instance features to facilitate instance segmentation. Meanwhile, this module also aggregates instance features into semantic feature space to promote semantic segmentation. Specifically, the semantic feature matrix F_{SS} is transformed into instance feature space as F_{SST} by a 1D convolution (Conv1D), and the F_{SST} is added to instance feature matrix F_{IS} element-wise as F_{ISS} . Then, we model the spatial correlation of point features to enhance important features by concatenating the feature F_{IS} and F_{ISS} into a F_{ISSC} , and then the F_{ISSC} is applied a mean of elements across dimension (Mean) and an element-wise sigmoid (Sigmoid) to generate a weight matrix F_{ISSR} . Finally, the feature matrix F_{ISSC} multiply the F_{ISSR} to generate the feature matrix F_{ISSSR} followed by two 1D convolution to produce the instance embedding feature E_{ISS} shaped with $N_a \times K$. The process can be formulated as follows:

$$F_{ISSC} = \text{Concat}(F_{IS}, F_{IS} + \text{Conv1D}(F_{SS})), \quad (5)$$

$$F_{ISSR} = F_{ISSC} \cdot \text{Sigmoid}(\text{Mean}(F_{ISSC})), \quad (6)$$

$$E_{ISS} = \text{Conv1D}(\text{Conv1D}(F_{ISSR})), \quad (7)$$

where instance embedding feature matrix E_{ISS} is used to generate final instance labels by using mean-shift clustering.

For the semantic segmentation branch, given the instance embeddings F_{ISSR} , this module integrates the F_{ISSR} into semantic feature space as F_{ISSST} with a 1D convolution followed by a mean of elements across dimension and a tiling operation. Next, other operations are similar to the instance branch except the last layer which outputs an instance-aware semantic feature matrix P_{SSI} shaped with $N_a \times C$. We also formulate this procedure as follows:

$$F_{ISSST} = \text{Tile}(\text{Mean}(\text{Conv1D}(F_{ISSR}))), \quad (8)$$

$$F_{SSI} = \text{Concat}(F_{SS}, F_{SS} + F_{ISSST}), \quad (9)$$

$$F_{SSIR} = F_{SSI} \cdot \text{Sigmoid}(\text{Mean}(F_{SSI})), \quad (10)$$

$$P_{SSI} = \text{Conv1D}(\text{Conv1D}(F_{SSIR})), \quad (11)$$

where F_{SSI} is a instance-fused feature matrix and the F_{SSIR} is a feature fusion matrix for semantic segmentation. The final instance-aware semantic features are fed into the last classifier to predict the categories for each point.

Experiments

Datasets and Evaluation Metrics

We evaluate our approach on the following two public datasets: Stanford Large-Scale 3D Indoor Spaces (S3DIS) (Armeni et al. 2016) and ShapeNet (Yi et al. 2016). The S3DIS is an indoor 3D point cloud dataset that contains six areas of three different buildings and have 272 rooms and involve 13 categories in total. For a principled evaluation, we follow the same k-fold cross validation as in (Qi et al. 2017a), and we also present the results of the 5-th fold (Area 5) following (Tchapmi et al. 2017) because Area 5 is not in the same building as other areas and there exist some differences between the objects in Area 5 and other areas. Moreover, we also evaluate our algorithm on ShapeNet dataset.

This dataset contains 16881 CAD models from 16 categories annotated with 50 types of parts and the models in each category are labeled with two to five parts. We follow the official split of 795 scenes as training set, 654 scenes as testing set. The instance annotations generated following (Wang et al. 2018a) are regarded as instance ground truth labels.

For semantic segmentation evaluation, overall accuracy (oAcc), mean accuracy (mAcc) and mean IoU (mIoU) are calculated across over all the categories. For instance segmentation, we evaluate our method including mean precision (mPrec), mean recall (mRec) with IoU threshold 0.5 and (weighted) coverage (Cov, WCov) (Ren and Zemel 2017; Wang et al. 2019b). The Cov scores measure the instance-wise IoU for each prediction matched with ground truth instance averaged over the scene. And then the Cov is further weighted with the size of ground-truth instances to obtain WCov. Given the predicted regions \mathcal{P} and ground truth regions \mathcal{G} , the Cov and WCov are formulated as:

$$\text{Cov}(\mathcal{G}, \mathcal{P}) = \sum_{m=1}^{|\mathcal{G}|} \frac{1}{|\mathcal{G}|} \max_n \text{IoU}(r_m^{\mathcal{G}}, r_n^{\mathcal{P}}), \quad (12)$$

$$\text{WCov}(\mathcal{G}, \mathcal{P}) = \sum_{m=1}^{|\mathcal{G}|} w_m \max_n \text{IoU}(r_m^{\mathcal{G}}, r_n^{\mathcal{P}}), \quad (13)$$

$$w_m = \frac{|r_m^{\mathcal{G}}|}{\sum_k |r_k^{\mathcal{G}}|}, \quad (14)$$

where $|r_m^{\mathcal{G}}|$ is the number of points in ground truth region m .

Implementation Details

For the large scale dataset S3DIS, each point in our model is represented by a 9-dim vector (XYZ, RGB and normalized location as to the room). Following experimental settings in PointNet (Qi et al. 2017a), we split the rooms into overlapped blocks of area $1m \times 1m$, and each block contains 4096 points. During training process, we configure the network with $\delta_v = 0.5$, $\delta_d = 1.5$ and $K = 5$, where K is the dimension of the embedding. We train the network for 100 epochs with batch size 24 on a single NVIDIA GTX1080Ti. We use Adam optimizer to optimize the network with momentum set to 0.9, base learning rate set to 0.001, and decay by 0.5 every 12.5k iterations. At test time, We use mean-shift clustering with bandwidth 0.6 to generate instance objects and merge instances of different blocks by using BlockMerging algorithm (Wang et al. 2018a). For ShapeNet dataset, each model is sampled into a point cloud with 2048 points represented by a 6-dim vector (XYZ and normal) as in (Qi et al. 2017b).

Instance Segmentation on the S3DIS dataset

As is depicted in Table 1, we present the performance of our approach in instance segmentation task on S3DIS dataset. In this task, we evaluate and compare our method with exist state-of-the-art methods including SGPN (Wang et al. 2018a), MT-PNet (Pham et al. 2019b), MV-CRF (Pham et al. 2019b), ASIS (Wang et al. 2019b), 3D-BoNet (Yang et

Table 1: Instance segmentation results on S3DIS dataset.

	Method	mCov	mWCov	mRec	mPrec
5-th fold	SGPN	32.7	35.5	28.7	36.0
	ASIS	44.6	47.8	42.4	55.3
	JSNet (Ours)	48.7	51.5	46.9	62.1
6 fold	SGPN	37.9	40.8	31.2	38.2
	MT-PNet	-	-	-	24.9
	MV-CRF	-	-	-	36.3
	ASIS	51.2	55.1	47.5	63.6
	3D-BoNet	-	-	47.6	65.6
	JSNet (Ours)	54.1	58.0	53.9	66.9

Table 2: Semantic segmentation results on S3DIS dataset.

	Method	mAcc	oAcc	mIoU
5-th fold	PointNet	52.1	83.5	43.4
	SEGCloud	57.4	-	48.9
	RSNet	59.4	-	51.9
	3P-RNN	71.3	85.7	53.4
	ASIS	60.9	86.9	53.4
	JSNet (Ours)	61.4	87.7	54.5
6 fold	PointNet	60.3	80.3	48.9
	3P-RNN	73.6	86.9	56.3
	MT-PNet	-	86.7	-
	MV-CRF	-	87.4	-
	ASIS	70.1	86.2	59.3
	JSNet (Ours)	71.7	88.7	61.7

al. 2019). We can see our network outperforms the other methods on S3DIS. Among them, ASIS is the most similar approach to our method. Compared with ASIS, our network JSNet achieves significant improvements on the four evaluation metrics. Especially on Area 5 of S3DIS, the improvements are more significant for each metric: 4.1 mCov, 3.7 mWCov, 4.5 mRec and 6.8 mPrec. Compared with the latest method 3D-BoNet on six fold experiments, our approach is also slightly better. Qualitative results are presented in Figure 3.

Semantic Segmentation on the S3DIS dataset

Table 2 presents the quantitative results of our architecture in semantic segmentation task on S3DIS dataset. As is seen from Table 2, our approach outperforms the baseline PointNet (Qi et al. 2017a) by 11.4 mAcc, 8.4 oAcc and 12.8 mIoU in over all accuracy on six fold cross validation experiments. For the generalizability evaluation on Area 5 of S3DIS, the performance is improved with 9.3 mAcc, 4.2 oAcc and 11.1 mIoU respectively. In addition, we also compare our method with other state-of-the-art methods on 6 fold or 5-th fold of S3DIS. Our model is slightly better than SEGCloud (Tchapmi et al. 2017), RSNet (Huang, Wang, and Neumann 2018), 3P-RNN (Ye et al. 2018), MT-PNet (Pham et al. 2019b), MV-CRF (Pham et al. 2019b), and ASIS (Wang et al. 2019b). Qualitative results are presented in Figure 4.

ShapeNet Results

Besides evaluation on the large scale indoor real scene benchmark S3DIS, we also conduct experiments on

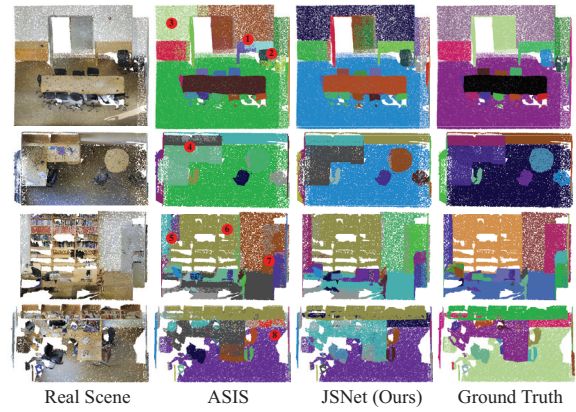


Figure 3: Comparison results of ASIS and our method in instance segmentation task on S3DIS. Different colors represent different instances. Different numbers indicate that the segmentation results of our method are better than the ASIS in nearby area.

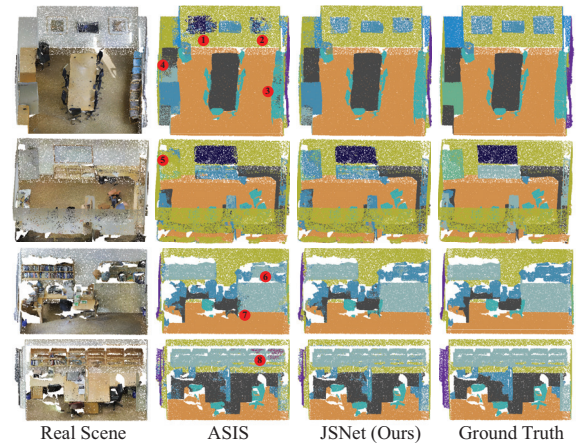


Figure 4: Comparison results of ASIS and our method in semantic segmentation task on S3DIS. Different numbers indicate that the segmentation results of our method are better than the ASIS in nearby area.

ShapeNet dataset. Following (Wang et al. 2018a), the instance annotations are generated as ground truth to train our network. Since these annotations are fake ground truth labels, we only present the qualitative results of part instance segmentation, as is illustrated in Figure 5. The results of semantic segmentation are reported in Table 3. We use PointNet++ (Qi et al. 2017b) as our baseline, and JSNet outperforms the baseline by 0.9-point mIoU. Compared with ASIS (Wang et al. 2019b), our approach also achieve an improvement of 0.8 mIoU. These results show that our approach is also favorable for the part segmentation task.

Ablation Study

To better validate the effectiveness of each component in our network, we conduct 7 groups of ablation experiments on Area 5 of S3DIS dataset. In addition, we also conduct addi-

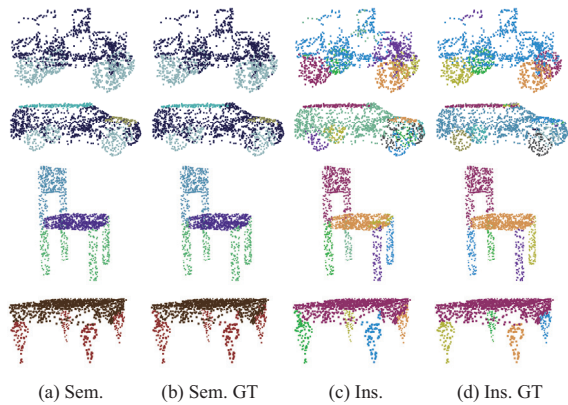


Figure 5: Qualitative results of JSNet on ShapeNet dataset. (a) Semantic prediction results of JSNet. (b) Ground truth of semantic segmentation. (c) Instance prediction results of JSNet. (d) Generated instance annotation for instance segmentation.

Table 3: Semantic segmentation results on ShapeNet dataset.

Method	mIoU
PointNet	83.7
PointNet++	84.9
ASIS	85.0
JSNet (Ours)	85.8

tional experiments to validate the effects of different training strategies on the same dataset. For all ablation experiments, if there are no extra notes, we use the same configuration in the subsection Implementation Details.

(1) **Base Network.** The base network includes a shared encoder and two parallel decoders. The encoder is built by stacking four set abstraction modules of PointNet++ (Qi et al. 2017b), and the decoders are built by stacking four feature propagation modules of PointNet++.

(2) **Backbone Network.** The encoder of backbone is built by concatenating a set abstraction module of PointNet++ and three feature encoding layers of PointConv (Wu, Qi, and Fuxin 2019). Similarly, the decoders are composed with three depthwise feature decoding layers of PointConv following a feature propagation module of PointNet++.

(3)-(6) **Single Module Evaluation.** We remove other components from the full framework (7) and only retain a module for the ablation experiments respectively.

(8)-(10) **Different strategies.** we train the full model with early stopping or random sample.

In Table 4, we present the ablation experimental results of different components in the full framework. Compared with the base network, the backbone network indeed benefits from a more efficient real convolution with density weighted. Compared with (2) and (3), the experimental results shows that fusing the feature of different layers could improve the segmentation precision because of the richer features after fusing. As for the only instance fusion semantic segmentation and only semantic awareness instance seg-

Table 4: Ablation experiments results on Area 5 of the S3DIS. The short names for components and strategies are defined as: BN—Base network, BBN—Backbone network, IF—Instance fusion branch of JISS module, SF—Semantic fusion branch of JISS module, ES—Early stopping, RS—Random sample.

Group	Component					Strategy		Metric	
	BN	BBN	PCFF	IF	SF	ES	RS	mPrec	mIoU
(1)	✓							52.3	52.7
(2)		✓						55.9	53.0
(3)		✓	✓					58.6	54.5
(4)		✓		✓				56.9	53.5
(5)		✓			✓			57.2	53.5
(6)		✓		✓	✓			58.6	54.3
(7)		✓	✓	✓	✓			57.6	54.3
(8)		✓	✓	✓	✓	✓		58.7	54.4
(9)		✓	✓	✓	✓		✓	62.1	54.5
(10)		✓	✓	✓	✓	✓	✓	62.9	55.0

mentation, the results indicate that better instance predictions could assign more reliable category labels to semantic branch, which can improve the performance of semantic segmentation. Similarly, the semantic awareness could enhance the instance predictions. In sixth ablation experiment, we combine instance fusion with semantic awareness, and the performance improvement is larger than only using one of them.

Table 4 also depicts the ablation experiments results of the full framework trained with different schemes including fix sample, fix sample with early stopping, random sample and random sample with early stopping. Compared with (7) and (8), our model has a slight overfitting and we alleviate this phenomenon by training the network with early stopping. Compared the fix sample (7) with random sample (9), the results indicates that we train the full framework with the strategies random sample, which makes our model avoid overfitting and has stronger generalization ability. In addition, we also use the early stopping strategies to train the network with random sample. our approach achieves 62.9 mPrec and 55.0 mIoU for instance segmentation and semantic segmentation respectively.

Conclusion

In this work, we propose JSNet, which is a novel end-to-end approach based on deep learning framework for 3D instance segmentation and semantic segmentation on point clouds. The framework consists of a shared feature encoder, two parallel feature decoders followed by a point cloud feature fusion (PCFF) module respectively and a joint instance semantic segmentation (JISS) module. On the one hand, the feature encoder, the feature decoders and the PCFF module can learn more effective and more discriminative features. On the other hand, the JISS module make the instance and semantic segmentation take advantage of each other. Finally, our approach achieves a significant improvement in both instance and semantic segmentation tasks on S3DIS dataset. In the future, spatial geometric topology of point clouds can be added into our framework for better segmentation results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61772213 and 91748204.

References

- Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *CVPR*.
- Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.
- Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI* (5):603–619.
- De Brabandere, B.; Neven, D.; and Van Gool, L. 2017. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.
- Hermosilla, P.; Ritschel, T.; Vázquez, P.-P.; Vinacua, À.; and Ropinski, T. 2018. Monte carlo convolution for learning on non-uniformly sampled point clouds. In *SIGGRAPH Asia 2018 Technical Papers*, 235. ACM.
- Huang, J., and You, S. 2016. Point cloud labeling using 3d convolutional neural network. In *ICPR*.
- Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; and Wang, X. Mask Scoring R-CNN. In *CVPR*.
- Huang, Q.; Wang, W.; and Neumann, U. 2018. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *NIPS*.
- Li, Y.; Chen, X.; Zhu, Z.; Xie, L.; Huang, G.; Du, D.; and Wang, X. 2019. Attention-guided unified network for panoptic segmentation. In *CVPR*.
- Li, J.; Chen, B. M.; and Hee Lee, G. 2018. So-net: Self-organizing network for point cloud analysis. In *CVPR*.
- Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*.
- Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.
- Maturana, D., and Scherer, S. 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*.
- Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019a. Real-time progressive 3d semantic segmentation for indoor scenes. In *WACV*.
- Pham, Q.-H.; Nguyen, T.; Hua, B.-S.; Roig, G.; and Yeung, S.-K. 2019b. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *CVPR*.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*.
- Ren, M., and Zemel, R. S. 2017. End-to-end instance segmentation with recurrent attention. In *CVPR*.
- Simonovsky, M., and Komodakis, N. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*.
- Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; and Savarese, S. 2017. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*.
- Thanh Nguyen, D.; Hua, B.-S.; Tran, K.; Pham, Q.-H.; and Yeung, S.-K. 2016. A field model for repairing 3d shapes. In *CVPR*.
- Wang, W.; Yu, R.; Huang, Q.; and Neumann, U. 2018a. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*.
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2018b. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*.
- Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019a. Graph attention convolution for point cloud semantic segmentation. In *CVPR*.
- Wang, X.; Liu, S.; Shen, X.; Shen, C.; and Jia, J. 2019b. Associatively segmenting instances and semantics in point clouds. In *CVPR*.
- Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*.
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*.
- Xu, Y.; Fan, T.; Xu, M.; Zeng, L.; and Qiao, Y. 2018. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *ECCV*.
- Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S.; Markham, A.; and Trigonì, N. 2019. Learning object bounding boxes for 3d instance segmentation on point clouds. *arXiv preprint arXiv:1906.01140*.
- Ye, X.; Li, J.; Huang, H.; Du, L.; and Zhang, X. 2018. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *ECCV*.
- Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; Guibas, L.; et al. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)* 35(6):210.
- Yi, L.; Zhao, W.; Wang, H.; Sung, M.; and Guibas, L. J. 2019. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*.
- Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018. Pu-net: Point cloud upsampling network. In *CVPR*.
- Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*.