Joint Semantic-Instance Segmentation Method for Intelligent Transportation System

Yujie Li, Jintong Cai, Quan Zhou, *Member, IEEE*, and Huimin Lu

***Abstract*— Getting the point cloud data from sensors and correctly understanding the scene is the core of the intelligent transportation system. Point cloud segmentation can help intelligent transportation systems distinguish different objects in the scene. Some methods process the point cloud through a feature extraction network and complete the segmentation task. However, these methods have high requirements on the feature extraction network, and the fineness of the features will directly affect the final segmentation result. In this paper, we propose a new feature extraction network for segmentation by adding an encoder-decoder structure, which can extract the multiscale local feature information from the feature map. In our opinion, the merged multiscale features obtain a better feature matrix, which improves the performance of the segmentation tasks. We report results on the S3DIS dataset, new feature extraction network greatly improves both semantic segmentation and instance segmentation tasks.**

***Index Terms*— Feature extraction, instance segmentation, multiscale fusion, semantic segmentation.**

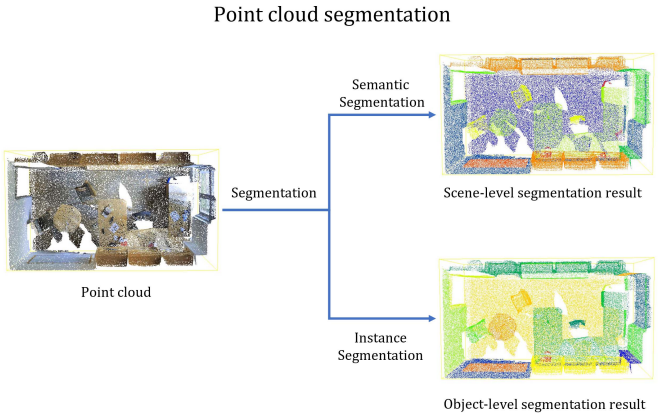

Fig. 1. Difference between semantic and instance segmentation. The result of semantic segmentation is scene-level, distinguishing only objects of different categories in the scene, such as tables and chairs. In object level instances, different objects of the same class are distinguished.

I. Introduction

THE intelligent transportation system [1], [28] is to effectively and comprehensively apply advanced technologies such as computer science and automatic control theory to transportation and vehicle manufacturing, and finally form a safe and efficient comprehensive transportation system [29]. As the most important part of an intelligent transportation system, how to obtain and express scene information with high reliability and accuracy is the first problem to be solved. In recent years, 3D acquisition technology has recently developed rapidly, using lidar to generate accurate point cloud data [2]. Point clouds can represent raw geometric information in 3D space as a simple and efficient data representation format. How to construct a mathematical model to represent, process, and analyze point cloud data has become an urgent problem to be solved.

Deep learning on the point cloud has wide universality. For various problems in this field, many methods have been proposed, such as segmentation [2], [3]. Point cloud segmentation refers to the accurate classification judgment of the scene point by point and depicts the boundary contour of the object, which can achieve higher-precision scene understanding. As shown in Fig. 1, 3D point cloud segmentation can be classified into semantic segmentation [4] or instance segmentation [5] according to the granularity of segmentation. Semantic segmentation is scene-level. The goal of semantic segmentation is to divide points into subsets according to their semantic classes. Instance segmentation is object-level because it requires more precise and fine-grained point inference. Therefore, instance segmentation is more challenging than semantic segmentation.

Currently, some approaches have begun to focus on the correlation between instance and semantic segmentation and consider combining them into one task [6]. Semantic segmentation and instance segmentation do have similarities. For example, each object in an instance segmentation has the same semantic label. In semantic segmentation, points with different semantic labels must not belong to the same object. Therefore, semantic segmentation first simplifies the task of instance segmentation. However, this segmentation method is implicit. It uses a single PointNet as a feature extraction

Manuscript received 17 January 2022; revised 20 March 2022 and 15 May 2022; accepted 22 June 2022. Date of publication 15 July 2022; date of current version 29 November 2023. The Associate Editor for this article was Z. Lv. *(Corresponding author: Huimin Lu.)*

Yujie Li is with the School of Information Engineering, Yangzhou University, Yangzhou 225012, China, and also with the School of Engineering, Kyushu Institute of Technology, Kitakyushu 804-8550, Japan (e-mail: yzyjli@gmail.com).

Jintong Cai is with the School of Information Engineering, Yangzhou University, Yangzhou 225012, China (e-mail: cjt794138599@qq.com).

Quan Zhou is with the National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210049, China (e-mail: quan.zhou@njupt.edu.cn).

Huimin Lu is with the Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Kitakyushu 804-8550, Japan (e-mail: luhuimin@ericlab.org).

Digital Object Identifier 10.1109/TITS.2022.3190369

network, and global features are directly obtained by max-pooling [7], which causes a loss of the local feature information. We find that fine-grained feature extractors greatly affect the performance of segmentation tasks [8], [9]. For the input point cloud, the larger the dimension of feature mapping space is, the easier it is to distinguish features. At the same time, sampling the point cloud and fusing the features of different layers can make the feature matrix more informative. Therefore, we believe that the structure of the feature extraction network will affect the representation ability of features, which in turn affects the segmentation performance. Designing a good feature extraction network to obtain multi-scale and richer features is the core focus of this method.

Therefore, referring to PointNet [7] and PointNet++ [8], we use the encoder-decoder structure to design a new feature extraction network. In the encoder, point clouds are grouped at different scales to form a local area, and multiple PointNets are used for feature extraction to obtain multiple feature maps of different sizes containing local information. In the decoder, we use the interpolation method to merge the information of the small-size feature layer into a large-size feature layer to output the final feature matrix. In summary, our study's main research contributions are as follows:

- A new feature extraction network is designed. Using an encoder that combines global features and local features to obtain multiscale and fine-grained feature layers.
- We design a new decoder with a fusion of multiscale feature layers as the final output, which is passed to the segmentation task.
- Instance segmentation and semantic segmentations are processed in parallel. The feature information of the two branches is merged to intensify the final effect of the two branches' tasks.

II. Related Work

In this section, we first introduce recent developments in semantic segmentation and instance segmentation. Finally, we analyze the importance of feature extraction network to point cloud analysis.

A. Semantic Segmentation on Point Cloud

Semantic segmentation divides all points in the scene into several subsets according to their semantic information. The current mainstream method is the point-based method [7], [8].

The typical method used is PointNet [7]. As the point cloud is disordered, PointNet proposes using symmetric function max-pooling to gather the information of each point. It also combines global features and local features so that the network can obtain richer feature information. However, PointNet has some limitations. It obtains fewer local features and does not have a multiscale feature extraction process. In response to the abovementioned shortcomings, PointNet++ [8] draws on the idea of a feature pyramid [10], which divides the point cloud into several local areas with overlapping parts, gradually expanding the extraction range and providing higher-level features. The entire network can be viewed as an encoder-decoder structure. The encoder is a sampling process, and multiscale features are obtained through multilayer set abstraction sampling. The decoder performs a reverse interpolation operation, the restored feature map contains the feature information of multiple scales, and finally, the semantic label of each point is obtained.

B. Instance Segmentation on Point Cloud

Instance segmentation needs to separate different objects of the same type from each other, which requires higher accuracy and fine granularity. There are two different types of instance segmentation methods, the proposal-based method [11], [12], and the proposal-free method [6], [13].

Proposal-based methods need to predict bounding boxes before segmentation. Yang *et al.* [11] proposed a point cloud instance segmentation network called 3D-BoNet. This method directly performs a rough 3D bounding box regression on all possible instances and then uses a point-level binary classifier to obtain the final instance label. The task of generating bounding boxes is expressed as an optimal allocation problem. The proposed multicriteria loss function is used to regularize the generated bounding boxes. This method is computationally efficient and does not require any postprocessing. In general, the proposed method is intuitive, and the results of instance segmentations usually have better visibility. The disadvantages of these methods are that they require multistage training and redundant proposals need to be deleted, which takes more time and is computationally expensive.

The proposal-free method does not need to calculate the bounding box of the object, but directly calculates the semantic features of the target, and then performs instance segmentation. Wang *et al.* [12] proposed a new method called Similarity Group Proposal Network (SGPN). According to the similarity metric learning principle, they believe that points belonging to the same instance should have similar features. Therefore, the model records all point-to-feature differences and generates a similarity matrix that measures whether each point belongs to the same instance. Some methods consider combining instance embeddings and semantic features. Wang *et al.* [13] proposed Associatively Segmenting Instances and Semantics (ASIS). Collecting semantic features into an instance feature matrix can generate semantically knowable instance embeddings. K-Nearest Neighbor (KNN) search [14] is used to ensure that adjacent points belong to the same instance that the semantic feature matrix containing the instances is obtained. The two segmentation branches are combined to finally output the instance segmentation result. Pham *et al.* [6] used the backbone network PointNet [7] for extracting the global feature matrix, and the proposed multi-task point-wise network (MT-PNet) was used to predict the class of 3D points, embedding 3D points in high dimensions. In the feature matrix, the points are clustered into instances and segmented using a multi-valued conditional random field (MV-CRF). In general, proposal-free methods are implicit, without obvious bounding boxes, but obtain distinguishable semantic features in high-dimensional space through feature extraction networks, which challenges the capabilities of feature extraction networks.

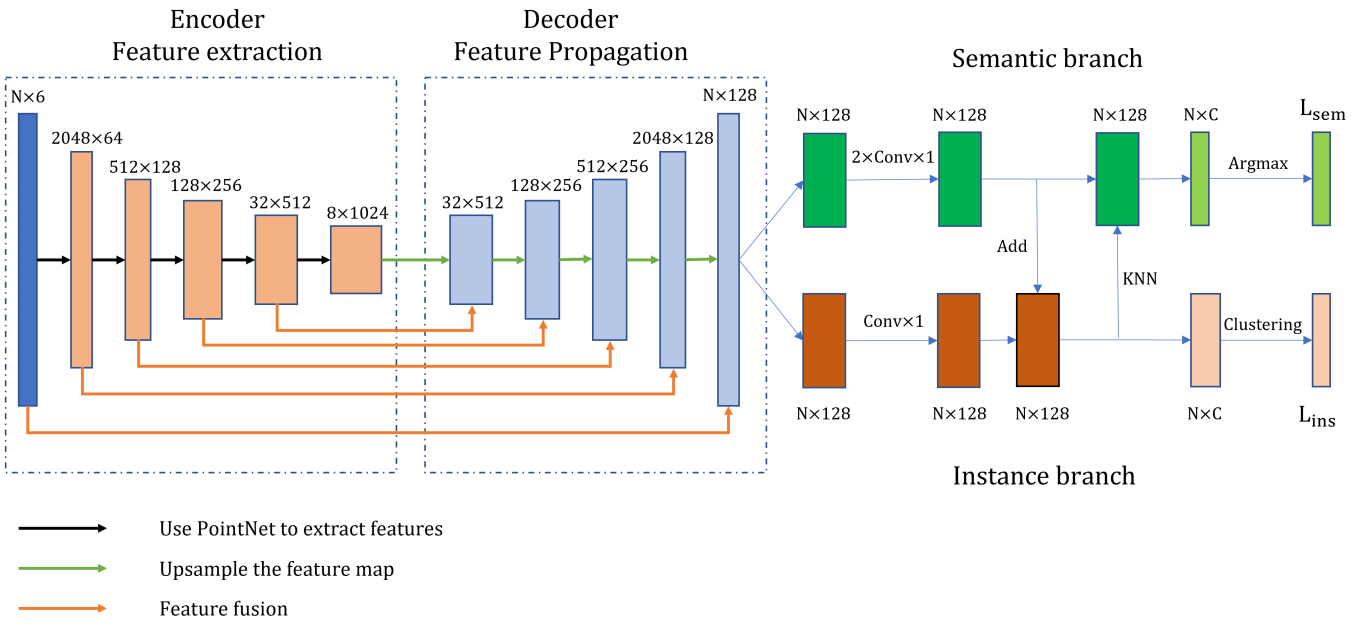

Fig. 2. Architecture of the proposed method. The encoder can obtain multi-scale feature layers, and the decoder can fuse the multi-scale feature layers, and the output is the same size as the input and contains richer features. The segmentation task is divided into semantic branch and instance branch. By adding semantic features to the instance features, get the instance embedding, and the final instance label can be obtained through clustering. Use KNN to set points belonging to the same instance, get the semantic feature matrix fused with instance, and use argmax to get the semantic label.

C. Feature Extraction Network on Point Cloud

PointNet uses Multi-Layer Perceptron (MLP)and global-pooling layers to build a general and scalable network for point cloud analysis. Since its feature extraction capability for local regions is still limited, recent research focuses on how to process features in local regions, that is, designing a more refined feature extraction network. There are three methods of local feature extraction: graph-based methods [15], convolution-based methods [16], and attention-based methods [19], [20].

Wang *et al.* [15] proposed a graph-based method that uses Edge Convolution (EdgeConv) to describe edge feature relationships between points and neighbors, captures local ensemble information, and guarantees feature invariance. Xu *et al.* [16] proposed to construct the convolution kernel by dynamically combining the basic weight matrix stored in the weight library. where the coefficients of these weight matrices are adaptively learned from point locations via ScoreNet [17]. It is more powerful and flexible than 2D convolution, and can better handle irregular and disordered point cloud data. Since Transformers [18] work well in the natural language processing (NLP) domain, some recent works focus on how to introduce attention mechanisms into point cloud processing models. Point Transformer [19] and Point Cloud Transformer [20] are representative works. For each point in the point cloud, the self-attention feature of the neighbor points is calculated, that is, the semantic relationship between each point is paid attention to, and the positional relationship between the points is also concerned. These methods have all designed fine local feature extractors, and they have shown good results in experiments. Therefore, we believe that the feature extraction structure is indispensable.

III. Method

Our method is shown in Fig. 2. To improve the accuracy of segmentation, we propose a joint semantic-instance segmentation network based on PointNet [7]. We design the feature extraction network as an encoder-decoder architecture. The input point clouds are extracted to feature layers of different scales through multiple PointNets. After that, the low-level features are interpolated and fused with a larger feature map. After 5 feature propagation operations, the output has the same size as the input and contains richer feature information. The segmentation task is divided into semantic branches and instance branches. Adding semantic features to the instance features can obtain the instance embedding incorporating the semantic features, and the final instance label can be obtained through clustering [21]. For the instance embedding space, the same instance points must belong to the same category, so the KNN [14] will be used to set the points belonging to the same instance to obtain the semantic feature matrix fused with the instance, and the semantic label is obtained through the argmax operation. Its main components are described in the following subsections.

A. Encoder Module

Finding a way to extract features from the input point cloud is key to our work. PointNet [7] can perform feature processing on disordered point clouds but it only focuses on global information. While point clouds are sparsely arranged in space, global features cannot show the characteristics of local dense points. Therefore, we need to reduce the receptive field and extract the features of the local area in the space.

We design the feature extraction process into an encoder structure, which is the core part of our feature extraction

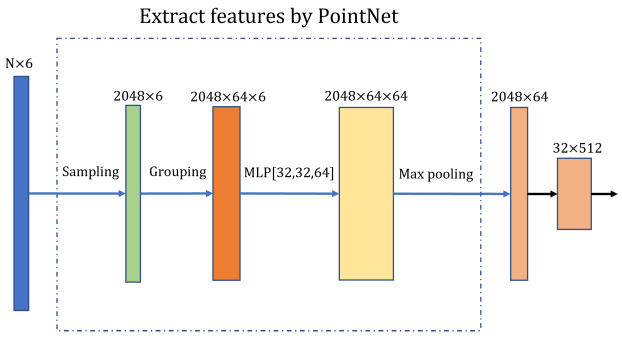

Fig. 3. Encoder module. We use multiple PointNet layers for feature extraction. Use the farthest point sampling on the input and group the processing results to form a local area. We use MLP and max pooling for feature extraction, and finally get a feature layer with a smaller size but richer feature information.

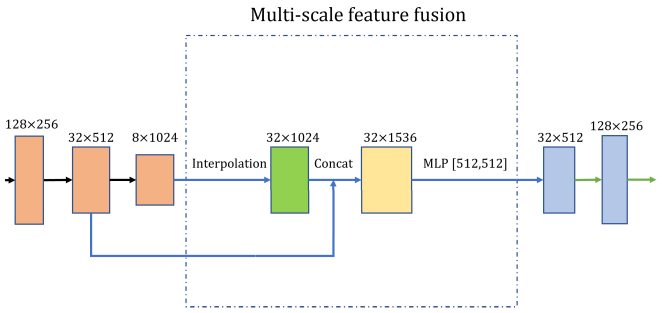

Fig. 4. Decoder module. We perform interpolation operations on the feature maps of small size, and concat with the higher feature map that are obtained by the decoder. The new feature maps contain details of different scales. Finally, we use MLP to adjust the feature channel.

network. In the encoder, we use multiple PointNet [7] layers for feature extraction. As shown in Fig. 3, the input feature is N × 6, and N is the number of sampling points. First, we use the farthest point sampling (FPS) [22] for the input and sample 2048 points out of N points. The layer size is 2048×6 after sampling. Since PointNet only treats the max-pooling feature as a local feature that loses too much detail, we refer to PointNet++ [8] to group the input feature layer and perform feature extraction for each group as a local feature. In our model, we select 64 neighbors for each point to form a group, and each group is equivalent to a local receptive field. After grouping, the size of the feature layer is $2048 \times 64 \times 6$. Then, we use the MLP to extract features for each channel and finally obtain a $2048 \times 64 \times 64$ feature map. We refer to the settings of PointNet and use max-pooling as a symmetric function to obtain a feature layer of 2048×64. Repeat the above steps, and finally get five sets of feature layers of different sizes, which are 2048×64, 512×128, 128×256, 32×512, and 8×1024.

Compared to PointNet, our encoder structure has multiple sampling layers of different sizes, and we perform feature extraction on multiple point groups in each sampling layer instead of a single point, which better describes the local features. The output of the encoder is transferred to the decoder for feature fusion operation.

B. Decoder Module

In the design of the feature extraction network, the main goal of the decoder is to fuse the features transmitted by the encoder, and finally generate a fine semantic feature matrix. How to use multiple feature layers in multiple encoders is our main purpose.

We refer to PointNet++ [8] to perform feature fusion on multiple feature layers of the decoder. The input point cloud is passed through the encoder, and 5 feature layers with different scales and rich information are obtained. Our decoder is shown in Fig. 4. We take two feature layers of 8×1024 and 32×512 as examples. First, an interpolation operation is performed on the 8×32 feature map, and the feature map is expanded to 32×1024. After that, we concatenate the 32×512 size feature map and the interpolated feature map. The new feature map contains details of different scales. Finally, we concat the result through MLP processing, and the size of the feature map after decoding is 32×512. We use the above method to decode each feature layer from the encoder and concatenate the multilayer feature information to finally obtain a feature layer with a size of N × 128. Finally, the output point cloud contains feature information of multiple scales, and the number of point clouds is the same as the input point cloud.

With a well-designed decoder, we reuse the multi-layer features obtained by the encoder and fuse the local information of different layers, and the final output feature matrix is more refined than PointNet. It avoids the direct segmentation task for each scale of the feature layer and simplifies the time and overhead of the segmentation task.

C. Joint Semantic-Instance Segmentation Module

Because of the powerful feature extraction network, we can process the obtained features and design the segmentation network. Our design refers to ASIS [13], which divides the feature output into two branches, semantic branches, and instance branches.

1) Instance Branch: The input of the instance branch is the feature matrix output by the feature extraction network, and our main goal is to perform object-level point-wise label prediction on the input point cloud.

First, the 1×1 kernels are performed twice on the semantic feature matrix. After the convolution operation, in the high-dimensional feature space, the points of different categories are separated, and the points of the same category will be placed closely together. We fuse new semantic features with instance features to obtain instance embeddings containing semantic information. Meanwhile, in the instance space, points belonging to instances of different categories are further separated, and instances of the same category will not be affected. In the inference stage, we use the mean-shift clustering [21] method to obtain the final instance label L_{ins}.

Since the final output of our feature extraction network is a feature matrix containing multi-scale information, we do not need to design complex multi-segmentation heads, which greatly simplifies the task of instance segmentation and reduces computational resource consumption. We make full use of semantic features to enrich our instance embeddings

without introducing too many post-processing operations, which are overall simple and efficient.

2) Semantic Branch: The goal of semantic segmentation is to perform scene-level point-wise label prediction. Please note that semantic segmentation only distinguishes points of different categories, and in the process of instance segmentation, the distinction of categories is inevitable, because points of different categories must not belong to the same instance. Therefore, we can enrich the semantic feature matrix with the help of instance embedding.

Similar to the operation of the instance branch, we perform two convolution operations on the output of the feature by the feature extraction network. After that, we use the KNN algorithm [14] to search for 64 neighbors for each point in the instance space, representing the feature information of the local area. We fuse the instance embedding containing neighbor features with the semantic feature matrix to produce a semantic feature matrix containing instance features. For the result of the semantic branching, we use the channel-wise maximum aggregation operation [23] to obtain the semantic label L_{sem} of each point.

We also abandoned the complex multi-segmentation head design, and only processed the feature matrix uniquely output by the feature extraction network to obtain the result of semantic segmentation. This also reflects that our feature extraction network can support complex segmentation tasks.

D. Loss Functions

Our loss function mainly supervises two branches, the semantic branch, and the instance branch.

1) Instance Branch Loss: The embedding of the internal pixels of the same instance should be as close as possible in the mapping space, and the average embedding vector of different instances should be as far away as possible. Therefore, we use discriminative loss [24] to supervise the instance branches. We design the loss function for instance segmentation as (1):

$$L_{INS} = \alpha \cdot L_{var} + \beta \cdot L_{dist} + \gamma \cdot L_{reg} \tag{1}$$

where:

$$L_{var} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}[\parallel \mu_c - x_i \parallel -\delta_V]_+^2 \tag{2}$$

$$L_{dist} = \frac{1}{C\,(C-1)}\sum_{c_A=1}^{C}\sum_{c_B=1}^{C}\left[2\delta_d - \parallel \mu_{c_A} - \mu_{c_B} \parallel\right]_+^2 \tag{3}$$

$$L_{reg} = \frac{1}{C}\sum_{c=1}^{C}\parallel \mu_c \parallel \tag{4}$$

$$\mu_c = \frac{1}{N}\sum_{i=1}^{N_c}\parallel x_i \parallel \tag{5}$$

where C represents the number of instances in the ground truth, and N_c in (5) represents the number of pixels in a certain instance. x_i represents the embedding vector generated by the i-th pixel in the instance and μ_c is the center of the embedding vectors corresponding to all pixels of the instance in the ground truth in the mapping space. C_A and C_B represent two different instances in space. δ_V and δ_d are margins. L_{var} in (2) pulls the instance embedding to the center of the instance, and L_{dist} in (3) making the embeddings of different instances move away from each other. L_{reg} in (4) is a regular item that makes the points of each cluster in the mapping space not too far away from the center. α, β, and γ adjust the weight of each part to L_{INS}.

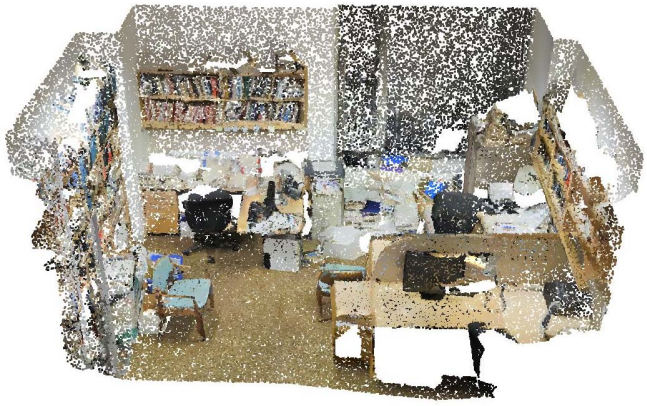

Fig. 5. Point cloud data in the S3DIS dataset. Objects in the scene are represented by colored points, and the goal of segmentation is to separate different objects in the scene.

2) Semantic Branch Loss: For the results of semantic segmentation, we use the cross-entropy loss [25] as (6):

$$L_{SEM} = -\sum_{i=1}^{N} y_i\, log\hat{y}_i + (1-y_i)\log\left(1-\hat{y}_i\right) \tag{6}$$

where y_i represents the prediction result of the i-th point, and $\hat{y}_i$ represents the label of the i -th point.

Finally, we design the total loss of the model as the sum of the semantic branch loss and the instance branch loss as (7):

$$Loss = L_{INS} + L_{SEM} \tag{7}$$

IV. Experiment

A. Dataset

We conducted experiments on the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [26], Fig. 5 shows a single scene in the dataset. The S3DIS dataset is an indoor dataset with pixel-level annotations developed by Stanford University. A Matterport camera was used to collect data and obtain the reconstructed 3D texture grid, original RGB-D image, and camera metadata of the scanned area and make a point cloud by sampling the grid. A semantic label was added to each point in the point cloud (for example, a total of 13 objects were used such as chairs, tables, floors, walls, etc.).

B. Training and Inference Details

Training Stage. For training, we referenced the PointNet [7] method, used the S3DIS dataset area 1, 2, 3, 4, 6 as train/valid, divided the room into small blocks of 1 m $\times$ 1 m, and performed sampling on each block. We reported our results on a single NVIDIA GEFORCE RTX 2080 GPU. The epoch

was 100, batch size was 8, the number of sampling points was 4096, the learning rate was 0.001, momentum was 0.9, the optimizer was SGD, decay step was 30000, and decay rate was 0.5. For the instance segmentation loss, we set it the same as in [24], $\alpha = \beta = 1$, $\gamma = 0.001$.

Inference stage. We used Area 5 as the test set to report our results. The instance segmentation branch used mean-shift clustering to obtain the instance label, and the bandwidth was 1. We used the argmax function of TensorFlow to obtain the final semantic label.

C. Evaluation Metrics

To evaluate the performance of semantic segmentation, we used overall accuracy (oAcc) and mean average classification accuracy (mAcc) [8] as evaluation metrics. oAcc represents the average accuracy of all test cases, and mAcc represents the average accuracy of all shape categories. In addition, we also used the mean intersection over union (mIoU) as evaluation metrics.

For example, for segmentation, we used mean class coverage (mCov) and mean class weighted coverage (WCov) [27] as evaluation metrics:

$$mCov\,(G, P) = \sum_{c=1}^{Class} \sum_{i=1}^{|G|} \frac{1}{G} \max_j IoU(r_i^G, r_j^P) \quad (8)$$

$$mWCov\,(G, P) = \sum_{C=1}^{Class} \sum_{i=1}^{|G|} w_i \max_j IoU(r_i^G, r_j^P) \quad (9)$$

$$w_i = \frac{|r_i^P|}{\sum_k r_i^G} \quad (10)$$

where C is the class number of the dataset, G represents the ground truth of each current instance and P represents the prediction of the current instance. mCov in (8) measures the maximum IoU of each ground truth instance of each class on the image. mWCov in (9) further weights the score by the size of the ground truth instance segmentation. w_i in (10) is the weight, which is related to the number of points in the instance. We also used the mean precision (mPrec) and mean recall (mRec).

D. Quantitative Results of Instance Segmentation

The results of the instance segmentation are shown in Table I. We compared with advanced SGPN [12] and ASIS [13] methods. SGPN is the first method to achieve point cloud instance segmentation. Our method is far ahead of SGPN in every metric, with mCov 16.0 higher than SGPN, 16.1 higher on mWcov, 20.7 higher on mPrec, and 16.4 higher on mRec, completely superior to the previous method. ASIS first proposed a method of combining semantic features with instance embedding to complete the task of point cloud segmentation, and our post-processing method is also similar to ASIS. Compared with the ASIS method, our method mRec is on par with ASIS, and other metrics are improved, mPrec is 1.3 higher.

By comparing with other methods, we can find that the refined feature extraction network can significantly improve the performance of instance segmentation tasks, and the multi-scale local features captured by the encoder-decoder structure are effective. Since the task of instance segmentation is more difficult, it has not reached saturation in various metrics, and subsequent improvements are still possible.

TABLE I
INSTANCE SEGMENTATION RESULTS ON THE S3DIS DATASET

Method	mCov	mWCov	mPrec	mRec
SGPN [12]	32.7	35.5	36.0	28.7
ASIS [13]	47.9	50.9	55.4	45.1
Ours	**48.7**	**51.6**	**56.7**	**45.1**

TABLE II
SEMANTIC SEGMENTATION RESULTS ON THE S3DIS DATASET

Method	mIoU	mAcc	oAcc
PointNet [7]	52.1	43.4	83.5
ASIS [13]	54.7	62.8	87.6
Ours	**55.7**	**63.2**	**87.9**

E. Quantitative Results of Semantic Segmentation

The semantic segmentation results are shown in Table II. We compare with PointNet [7] and ASIS [13].PointNet proposes a feature processing network that directly consumes point clouds, and our method also refers to PointNet for local feature processing. Compared with PointNet, our method leads PointNet comprehensively, with 3.6 higher on mIoU, 19.8 higher on mAcc, and 4.4 higher on oAcc. Compared with the semantic segmentation results of ASIS, our method also improves, where mIoU is 1.0 higher.

Since no object-level cooking point prediction is required, the semantic segmentation task is less difficult than the instance segmentation task, while the performance metrics are also higher. We can well integrate semantic information in high-dimensional space through the encoder-decoder structure, and adding instance embeddings to the semantic feature matrix also improves the performance of semantic segmentation tasks.

F. Qualitative Results

We present the segmentation results on the S3DIS dataset in Fig. 6. Column 2 and Column 3 show the ground truth and prediction results of semantic segmentation. Column 4 and Column 5 show the ground truth of instance segmentation. In addition, the predicted results are shown. Each row is a representative scene in the S3DIS dataset, and we report the results for 4 scenes to demonstrate the performance of our method.

In the semantic segmentation task, our task shows good results in all 4 scenes. In row 1, the predicted results of semantic segmentation (row 1, col 3) successfully classify tables and chairs in the scene. Note that semantic segmentation

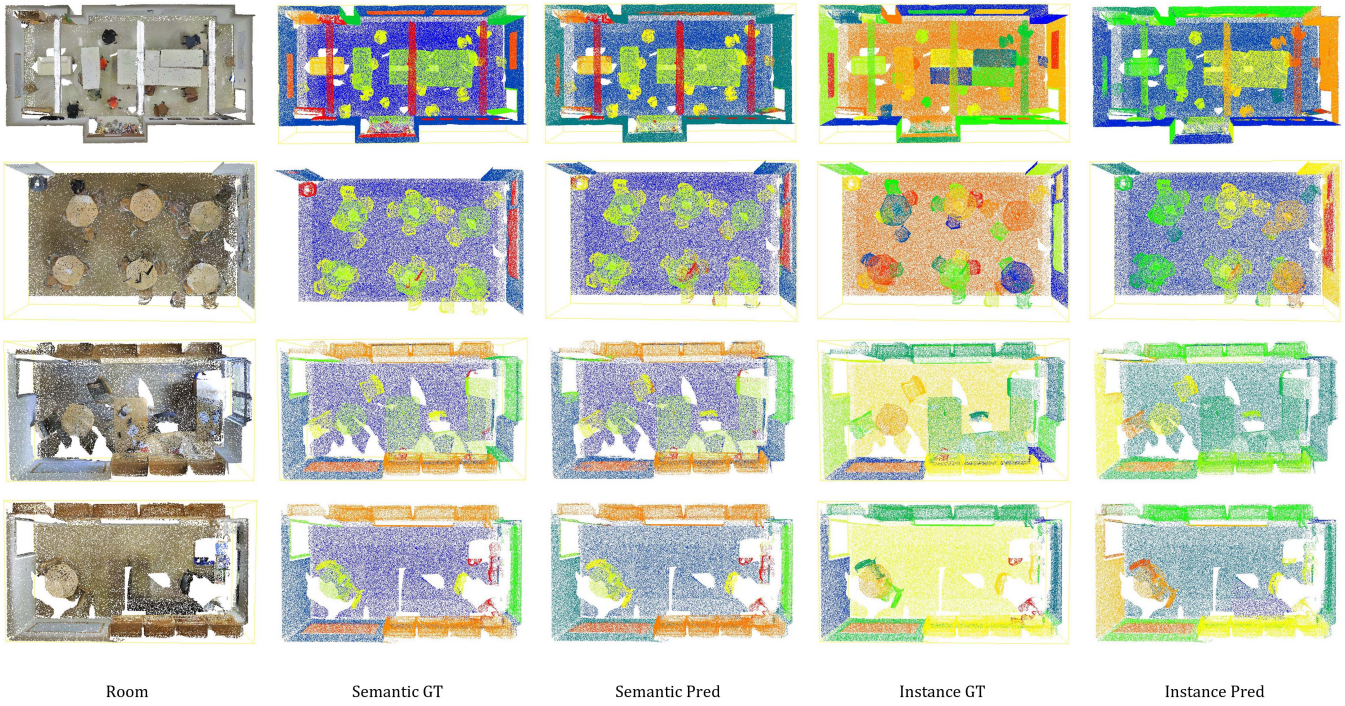

Fig. 6. Qualitative results on the S3DIS dataset. We give the visualization results of some rooms. Column 1 is the original point cloud. Column 2 and column 3 show the ground truth and prediction results of semantic segmentation. Column 4 and column 5 show the ground truth of instance segmentation. And predicted results.

only distinguishes categories, so the colors of the tables in the scene are the same. In row 2, the distance between multiple tables and chairs is very close, and the prediction result (row 2, col 3) is also correctly predicted. This proves that our model can accomplish the task of semantic segmentation. However, we also notice that there are mistakes in some places, in row 1, the window on the right side of the room is the same color as the wall, and the distance gap between the points is smaller. For this extreme case, the model does not correctly segment windows and walls (row1, col 3).

In the instance segmentation task, our model performs well in most cases. In row 3, our prediction results (row 3, col 5) can correctly classify the chair and table on the left side of the scene, and each chair has a different color, representing a different instance. In row 4, the predicted result (row 4, col 5) successfully distinguishes the bookcase from the floor. However, since the instance segmentation task is more complex, the accuracy of the overall task is still not high. In row 1, instance segmentation does not correctly divide the tables into 4 different instances (row1, col5). We believe that it is difficult for instance segmentation to distinguish the boundary points of point clouds when the instances in the scene are too close, so it is necessary to explore the method of boundary point distinction in instance segmentation in the future.

V. Conclusion

In this work, we analyze the necessity of designing a feature extraction network in point cloud processing tasks. Therefore, we design an encoder-decoder structure to process point clouds directly, the encoder generates rich multi-layer local features, and the decoder fuses the above features. Finally, we fuse instance embeddings with semantic features to complete both semantic segmentation and instance segmentation tasks. We present quantitative and qualitative results on the S3DIS dataset, which demonstrate the effectiveness of our method. At the same time, we find that our model still has room for improvement, when the objects in the space are close together, our model segmentation does not perform well on boundary points. In the future, we will explore ways to distinguish boundary points in segmentation tasks, design better models, and contribute to breakthroughs in autonomous driving.

References

[1] A. Belhadi, Y. Djenouri, G. Srivastava, and J. C.-W. Lin, "SS-ITS: Secure scalable intelligent transportation systems," *J. Supercomput.*, vol. 77, no. 7, pp. 7253–7269, Jul. 2021.

[2] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[3] T. Wang, Q. Yang, X. Shen, T. R. Gadekallu, W. Wang, and K. Dev, "A privacy-enhanced retrieval technology for the cloud-assisted Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4981–4989, Jul. 2022.

[4] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, Feb. 2019.

[5] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: State of the art," *Int. J. Multimedia Inf. Retr.*, vol. 9, pp. 171–189, 2020.

[6] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8827–8836.

[7] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017, *arXiv:1706.02413*.

[9] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano, and J. C.-W. Lin, "A two-phase anomaly detection model for secure intelligent transportation ride-hailing trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4496–4506, Jul. 2021.

[10] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[11] B. Yang *et al.*, "Learning object bounding boxes for 3D instance segmentation on point clouds," 2019, *arXiv:1906.01140*.

[12] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578.

[13] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4096–4105.

[14] V. Garcia, E. Debreuve, and M. Barlaud, "Fast K nearest neighbor search using GPU," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–6.

[15] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.

[16] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3173–3182.

[17] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4867–4876.

[18] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[19] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[20] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[21] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1197–1203.

[22] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling for implicit surfaces and point clouds," Comput. Lab., Univ. Cambridge, Cambridge, U.K., Tech. Rep. 565, 2003, pp. 1–12.

[23] M. Abdel-Nasser, A. Saleh, and D. Puig, "Channel-wise aggregation with self-correction mechanism for multi-center multi-organ nuclei segmentation in whole slide imaging," in *Proc. 15th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2020, pp. 466–473.

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

[25] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–11.

[26] I. Armeni *et al.*, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.

[27] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6656–6664.

[28] S. Pandya, T. R. Gadekallu, P. K. Reddy, W. Wang, and M. Alazab, "InfusedHeart: A novel knowledge-infused learning framework for diagnosis of cardiovascular events," *IEEE Trans. Computat. Social Syst.*, pp. 1–10, 2022.

[29] Y. Chen, X. Xu, and W. Wang, "Efficient web APIs recommendation with privacy-preservation for mobile app development in industry 4.0," *Trans. Ind. Inform.*, p. 1, 2021.