A Click-Based Interactive Segmentation Network for Point Clouds

Wentao Sun, Zhipeng Luo, *Member, IEEE*, Yiping Chen, *Senior Member, IEEE*, Huxiong Li, José Marcato Junior, *Member, IEEE*, Wesley Nunes Gonéalves, *Member, IEEE*, and Jonathan Li, *Fellow, IEEE*

***Abstract*— Interactive segmentation plays an essential role in several tasks involving point clouds. However, existing methods suffer from low segmentation accuracy and cannot adjust the segmentation results according to the user's personal demands. This article presents a novel deep-learning (DL)-based interactive segmentation method, named click rough segmentation network (CRSNet), designed to handle point clouds. The method allows users to iteratively click to segment interesting objects. CRSNet consists of two key parts: a click rough segmentation (CRS) module and a feature extraction module. First, the CRS module transforms click operations into an appropriate representation to input into the feature extraction module. The CRS module takes raw point clouds and clicks operations as input and outputs 3-D Gaussian vectors and roughly segmented blocks, which adapt to different-sized and densely distributed objects in complex environments. Second, the feature extraction module, which uses a novel mix loss-based analysis algorithm, extracts deep features and obtains instance segmentation results. The module is highly compatible because its backbones can be replaced by different DL architectures. Experimental results on the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI), Apolloscape, Roadmarking, Scannet, and SemanticKITTI datasets show that our method outperforms state-of-the-art semantic segmentation methods with one click. Moreover, our method can generalize well to unseen objects and datasets.**

***Index Terms*— Deep learning (DL), instance segmentation, interactive segmentation, point clouds.**

I. Introduction

INTERACTIVE segmentation has become one of the most popular research topics in recent years [1], [2], [3]. Its goal is to obtain segmented objects of interest from interactive information like strokes [4], [5], bounding boxes [6], [7], [8], [13], [15], and clicks [1], [2], [3]. These user-selected segmentation results can be used in various downstream tasks, such as medical object detection [9], [10], and annotation of images and videos [11].

In the field of point clouds, the cost of point cloud annotation is higher than that of images, significantly increasing the expenses associated with high-precision 3-D map production and other downstream tasks. Moreover, in recent years, there has been a substantial surge in point cloud data due to the widespread use of point cloud mapping equipment and depth cameras. However, traditional labeling methods are incapable of coping with this rapid expansion in point cloud data, presenting challenges in training large network models. As a result, there is an urgent need for more intelligent point cloud interactive segmentation algorithms to streamline the labeling process and reduce costs.

Traditionally, interactive segmentation of raw point clouds has been mainly accomplished by graph-cut-based algorithms with strokes or bounding boxes as the primary interactive behavior. Several examples can be found in literature, including the Max Flow Min Cut-based algorithm [14], the superpixels and graph-cut segmentation method [12], the boundary-aware and Markov random field-based method [13], the interactive foreground extraction tool [16], and two-scale graphs methods including the region-based graph and the pixel-level graph [17]. However, these graph-cut-based methods are only able to capture the low-level geometric characteristics of point clouds and inevitably lose most of the high-level geometry information. Moreover, the strokes or bounding box behaviors are often inconvenient in complex scenes. Therefore, the performance of these existing traditional methods is still far from satisfactory. In recent years, deep learning (DL) [18] has made great progress in point cloud processing. Several effective DL-based methods [19], [20], [21], [22], [23], [24] have been proposed to capture deep features in point clouds and achieve excellent performance. However, no click and DL-based interactive segmentation methods have been used for point clouds. This is due to the challenges of interactive segmentation, which can be summarized as follows.

First, it is difficult to obtain enough information for segmentation through simple clicks, unlike bounding boxes and strokes, which can easily locate the exact objects of interest due to the accurate geometry information provided by point clouds.

Manuscript received 3 December 2022; revised 22 April 2023 and 8 August 2023; accepted 23 September 2023. Date of publication 13 October 2023; date of current version 27 October 2023. This work was supported by the National Natural Science Foundation of China under Grant 42371343. *(Wentao Sun and Zhipeng Luo contributed equally to this work.) (Corresponding authors: Yiping Chen; Jonathan Li.)*

Wentao Sun and Huxiong Li are with the Institute of Artificial Intelligence and the Department of Computer Science and Engineering, School of Mechanical and Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang 312000, China (e-mail: hbycswt@gmail.com; jsj_lhx@126.com).

Zhipeng Luo is with the School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China (e-mail: zpluo@stu.xmu.edu.cn).

Yiping Chen is with the School of Geospatial Engineering and Science, Sun Yat-sen University, Zhuhai, Guangdong 519082, China (e-mail: chenyp79@mail.sysu.edu.cn).

José Marcato Junior and Wesley Nunes Gonéalves are with the Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande 79070-900, Brazil (e-mail: jose.marcato@ufms.br; wesley.goncalves@ufms.br).

Jonathan Li is with the Department of Geography and Environmental Management and the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Digital Object Identifier 10.1109/TGRS.2023.3323735

Second, DL-based methods require a large amount of data, but currently, public instance segmentation datasets for point clouds are scarcer compared to the image domain. Additionally, point clouds can be captured by different sensors and have varying point density and occlusion, making it challenging to design a practical method that can adapt to different sensors.

Third, there is a lack of mature and perfect network backbones for point clouds compared to the image domain. Although some networks that directly process points have been proposed in recent years, it is still difficult to meet the demands of interactive segmentation, which requires high running speed, high accuracy, and a large perception domain simultaneously.

We propose a novel click rough segmentation network (CRSNet) for interactive segmentation that extracts instance labels from point clouds and addresses the three challenges mentioned above. To overcome the first challenge, we drew inspiration from successful interactive image segmentation methods [1], [2], [3] and adopted clicks as the interactive action, which is a simple and time-saving option. To obtain enough spatial information, we convert the click action on point clouds into Gaussian vectors that provide the position and local information of the target objects.

To address the second challenge, we collected the mainstream public datasets that contain instance label information and augmented the data with some techniques to ensure thorough training and testing of our method. To obtain interactive information, we use the computer to randomly generate clicks for each object.

Regarding the third challenge, the architecture of our method can use different backbones, enabling us to update it easily. We also designed a different scale rough segmentation algorithm and a mixed loss to increase the perception domain without sacrificing speed and solve the scale problem of different-sized objects. Fig. 1 shows the difference between the manual extraction method and our proposed method, highlighting how the latter simplifies the extraction process.

Our proposed method was tested on various datasets, and experimental results demonstrate its ability to accurately segment objects of interest, outperforming state-of-the-art semantic segmentation methods while also generalizing well to unseen objects and datasets. The three main contributions of our work are as follows.

1) We introduce a novel interactive segmentation network, CRSNet, for point clouds that provides reliable instance results and works online. This method can generalize well to segment unseen objects and datasets.
2) An efficient interactive method and information fusion way are proposed. Only a few clicks are needed to quickly and accurately identify the target object from the surrounding objects. Additionally, users can iteratively click on objects to further correct the segmented results. With just one click, the network's segmentation performance surpasses that of state-of-the-art semantic segmentation methods.
3) We propose an efficient feature representation and analysis step with a novel mixed loss. The entire analysis step is highly compatible, allowing backbones to be replaced by different DL architectures.

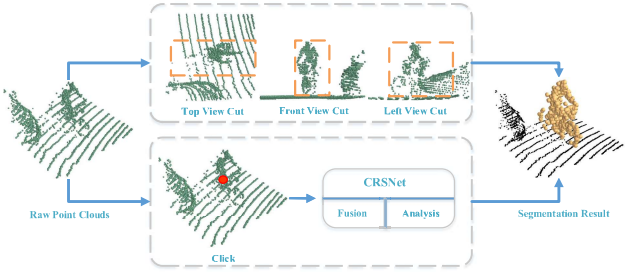

Fig. 1. Flowchart of manual extraction and CRSNet extraction methods. The upper part shows the manual extraction method, while the lower part demonstrates the CRSNet method, which comprises fusion and analysis steps. The red dot represents the click, the yellow points represent the object, and the black points represent the background.

II. Related Work

In the past few decades, numerous studies have been conducted in the field of interactive segmentation. This section provides a review of the significant works in both image and point cloud areas, as well as DL methods for point cloud segmentation.

A. Works in the Image Area

Interactive segmentation methods in the image area can be broadly categorized into two classes: hand-crafted feature-based methods and DL-based methods.

1) Hand-Crafted Feature-Based Methods: In the early years, most methods were based on hand-crafted feature-based methods. Initially, methods focused on boundary properties [25]. Boykov and Jolly [26] proposed a graphical-based model. Graph cut-based methods gained popularity, and several works were presented, such as the well-known min-cut/max-flow algorithms [27] and GrabCut [28]. Later, various tips were proposed to improve graph-cut-based methods [29], [30]. However, these methods based on low-level features fail to achieve perfect object segmentation results in complex environments.

2) DL-Based Methods: In contrast to traditional methods, DL methods can extract high-level features from images, significantly enhancing the final segmentation performance. Xu et al. [1] first proposed a convolutional neural network (CNN)-based network to address interactive segmentation, which employed positive and negative clicks as interactive inputs. Regional interactive image segmentation networks (RIS-Net) [31] considered local–regional information and used multiscale global contextual information to improve segmentation performance. Various other networks have also been proposed [2], [3], [32].

B. Works in Point Clouds

At present, there is no typical DL-based interactive segmentation method that allows users to constantly correct the object segmentation result by iteratively clicking or other behaviors. Existing interactive methods related to point clouds are mainly based on graph-cut methods and clustering methods.

1) Graph-Cut-Based Methods: Graph-cut-based methods aim to compute the minimum cut, which can partition the graph into two different parts. Transferring the point clouds into a graph is a vital process in these algorithms. Li [40] proposed one graph-cut-based method. It treats each point as a node and defines an edge weight function. Strokes are given to change the weight of each edge; finally, a minimum cut is obtained. However, this method is difficult to apply in other complicated environments since it distinguishes between urban buildings and trees. Later, Liu and Boehm [14] proposed a more general graph-cut-based method. Given the strokes, it can build a weighted graph, and the minimum cut problem can be solved by using the Max Flow Min Cut algorithm. Compared with the last method, this only needs two strokes and can potentially be applied to general point clouds. Luo et al. [13] further proposed one graph-cut-based method, which uses a boundary-aware Markov random field model to consider object-boundary constraints and achieve good results, but it requires bounding box inputs. Later, Luo et al. [15] updated their method and replaced the original graph-cut method with a graphical neural network, achieving good results. However, the input is complex, and users cannot constantly correct the segmentation results by interactive behaviors. In summary, for these graph-cut-based methods, the segmentation quality depends on the graph building, but the graph building process is difficult to adapt to different scenes.

2) Traditional Clustering Methods: Clustering methods are relevant to interactive segmentation because they can divide the points into different groups, and with the help of interactive behaviors, objects of interest are easily classified into the same group. Traditional clustering methods include k-means [39], k-means++ [34], mean-shift [35], seeded region growing [33], and so on. They use low-level features instead of understanding the geometric shape of the target in space to make judgments, which limits their effectiveness.

C. DL for 3-D Point Cloud Segmentation

3-D point cloud segmentation related to interactive segmentation can be classified into three categories: semantic segmentation, instance segmentation, and weakly supervised segmentation. All of these segmentation methods can help people quickly segment new point cloud scenes. A complete review of DL for 3-D point clouds is shown in [37].

1) DL for 3-D Semantic Segmentation: There are four kinds of methods: projection-based, discretization-based, point-based, and hybrid methods. Point-based methods are the most relevant to the interactive segmentation task because they can directly work on irregular point clouds without losing any geometric information. PointNet [19] is the first network to process raw point clouds directly. Its structure is simple and has a good ability to extract deep features, achieving good results in semantic segmentation and classification. However, it cannot capture the local information in point clouds. Later, PointNet++ [20], dynamic graph CNN (DGCNN) [21], and other new point-based networks [22], [23], [24] have solved this problem and have better performance than PointNet.

2) DL for 3-D Instance Segmentation: Instance segmentation is more challenging than semantic segmentation because it needs to distinguish the semantic meaning of each point and separate different instances simultaneously. Existing methods can be classified into two groups: proposal methods with the proposal module and proposal-free methods that aim to cluster instances after semantic segmentation. Famous proposal-based methods have a 3D-SIS network [38] that works on red, green and blue (RGB)-D images. Its structure is similar to 3-D object detection in the image area because it has a 3-D Regional Proposal Network and a 3-D Region of Interest module. Proposal-free methods do not have an object detection module. The representative work is SGPN [41], which introduces a similarity matrix to measure the similarity between feature pairs obtained from each point.

3) DL for Weakly/Semisupervised Segmentation: Unlike interactive segmentation, which can significantly reduce the cost of labeling point clouds, weakly/semisupervised methods allow the network to use much less labeled data during training. One representative work is [47], which uses learning gradient approximation and additional spatial and color smoothness constraints to reduce the need for training data by ten times. Wei et al. [48] introduced a novel method that introduces a multipath region mining module to generate pseudo-point-level labels from a classification network trained with weak labels. The pioneering work is a multiple instance learning (MIL)-derived transformer [49], which uses the transformer to explore pairwise cloud-level supervision. However, these methods aim to reduce the amount of data needed to train the network rather than reducing the labeling cost directly.

In summary, whether in images or point clouds, graph-cut-based algorithms, clustering methods, and other conventional methods all focus on low-level features such as density and connectivity. They give less consideration to the shape of objects, which can lead to a deterioration in more complex environments. Most DL-based segmentation methods can automatically segment point clouds, such as semantic segmentation, instance segmentation, and weakly/semisupervised segmentation. However, these methods cannot allow users to continuously correct the segmentation results by interacting with the model iteratively.

III. Proposed Method

In this section, we will introduce the problem definition and describe the architecture of our proposed method, which can be divided into two steps: fusion and analysis. The overall structure of the CRSNet is shown in Fig. 2.

A. Problem Definition

Our goal is to provide the instance label of the target in 3-D point clouds by clicks. The input is a set of points $P = \{p_i \,|i = 1, \ldots, n\}$ and clicks $C = \{c_i \,|i = 1, \ldots, n_c\}$, which consist of both positive and negative clicks. The final segmentation is represented by the label $\widetilde{L} = \{\widetilde{l_i}|i = 1, \ldots, n\}$ and a score k, where $\widetilde{l_i} \in (0, 1)$. A value of 0 means that the point p_i belongs to the background area, while a value of 1 means that the point p_i belongs to the target area. The value k represents the category of the object.

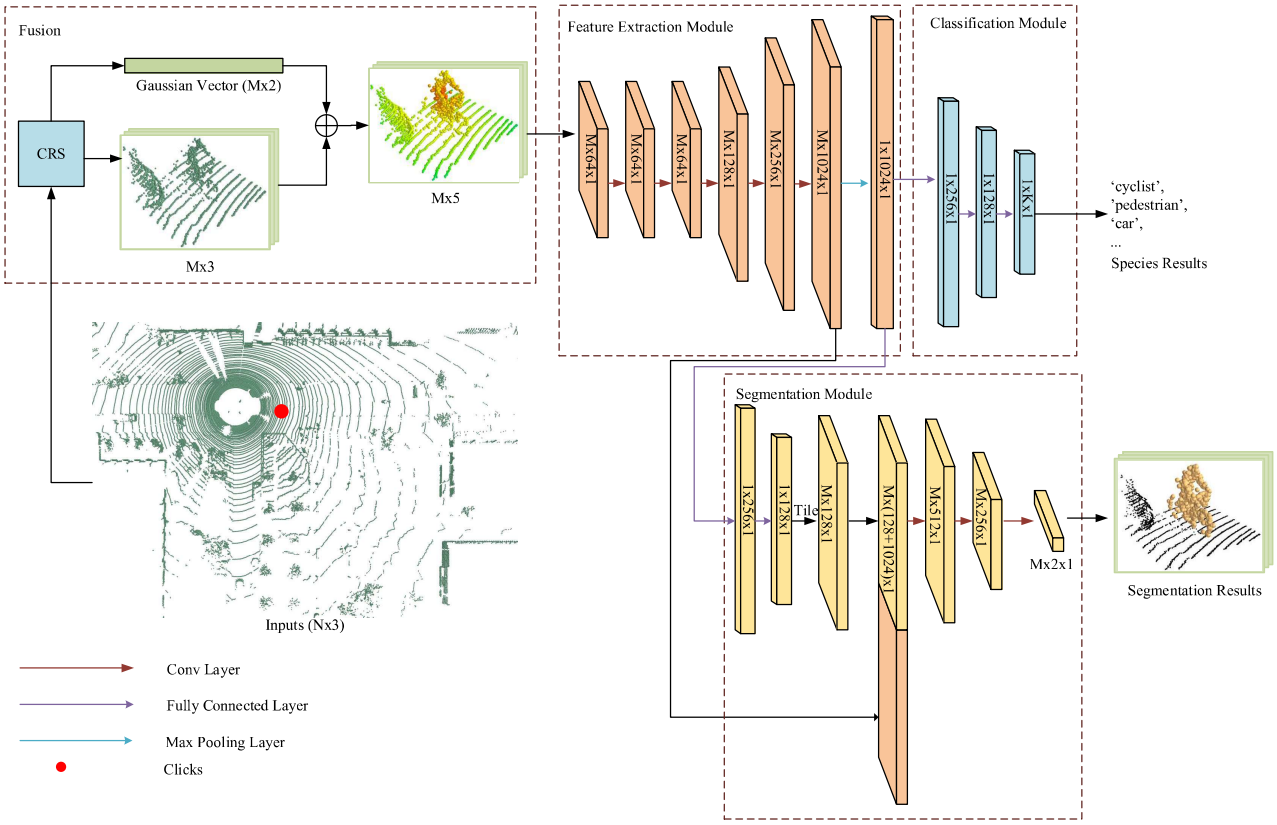

Fig. 2. Overview of the proposed CRSNet. K represents the number of object categories, N represents the number of points in the raw point clouds, and M represents the number of points in one rough segmentation block ($N > M$).

B. CRSNet Architecture

The structure of our proposed method is illustrated in Fig. 2. The CRSNet consists of two steps: the CRS module (fusion) and networks (analysis). In the fusion step, we input the click information and raw point clouds into the CRS module and obtain two Gaussian vectors (Mx2) and the rough segmentation blocks (Mx3), which are combined to form a series of CRS-Blocks (Mx5). In the analysis step, the CRS-Blocks are processed by networks to obtain the segmentation results and species results.

1) Fusion: The fusion step is a vital component of our proposed method. It transfers the raw point clouds and clicks actions into a suitable form that can be efficiently processed in the later analysis step. Fig. 2 illustrates that the fusion step comprises the CRS and aggregation operation modules.

a) CRS module: The CRS module is designed to solve the problems encountered when dealing with clicks and large amounts of raw point clouds. Clicks cannot be processed directly by a later network due to their format, and raw point clouds cannot be processed directly by a later network for two reasons: the network cannot deal with large amounts of points directly, and the network can only take a fixed number of points at once, so a sampling process is needed after the rough segmentation.

To overcome these challenges, we developed a click rough segmentation (CRS) module that consists of two steps: rough segmentation for point clouds and click transformation. First, the user clicks on the target object, and then a suitable-sized block centered on the click is obtained. We extract the block that contains the object of interest by a top view cut, and it has a fixed length and width. The scale of the block is set by the user, and even if the block size cannot match the object size precisely, we can still partition it into several parts and put them into different blocks.

Then, we convert the click position and scale size into a Gaussian vector. The whole CRS module can be defined mathematically, as shown in the equations below. Initially, a rough segmentation Pc is obtained by keeping only points close to the clicks C [see (1)–(3)]. Then, we calculate the Gaussian vector [see (4)]. As there are two different kinds of clicks: positive clicks and negative clicks, we finally can get two different Gaussian vectors

$$P = \{p_i | i = 1, \ldots, N\} \tag{1}$$

$$P_j = \left\{p_i \middle| \left\|p_i - c_j\right\| \le \sigma_s\right\} \subseteq P \tag{2}$$

$$Pc = \left\{P_1 \cup P_2, \ldots, \cup P_{N_c}\right\} \subseteq P \tag{3}$$

$$Gc = \left\{\sum_{j=1}^{N_c} \exp\left(-\left\|p_i - c_j\right\|^2 / 2\sigma_c^2\right) \middle| i = 1, \ldots, N_2\right\} \tag{4}$$

where P represents the raw point clouds and contains N points. P_j represents the segmented points according to the click c_j. σ_s represents the block size. P_c represents the rough segmented block clicked by all N_c clicks and has N_2 points. σ_c is the parameter that relates to the click size, and Gc is a Gaussian vector.

b) Aggregation operation for CRS-blocks: The inputs of this operation are Gaussian vectors and rough segmentation blocks. The Gaussian vectors can be taken as a new channel

of point clouds, so we concatenate the two inputs to form new blocks, named CRS-Blocks, with five channels (x, y, z, and two Gaussian vectors). CRS-Blocks contain both the click information and the object's shape information and can be processed more conveniently than the original two inputs.

2) Analysis: This step is a lightweight network that can simultaneously segment and classify the blocks generated by the fusion step. Fig. 2 shows that this step is composed of three modules: feature extraction module, segmentation module, and classification module. The whole network can be realized by different backbones, but we use PointNet as the backbone because of its remarkable running speed and clear structure, although it has poorer performance compared with other new networks.

a) Feature extraction module: It is designed to extract the features of point clouds. It comprises two parts: Conv mlp layers and a max-pooling layer. We input a series of blocks generated by the fusion module to the Conv mlp layers and get the feature map. Then, passing through the max-pooling layer, we extract a global feature map that describes the main characteristics of the point cloud.

b) Segmentation module: It is proposed to realize semantic segmentation by classifying each point into two categories: object and background. It first concatenates the feature map before the max-pooling layer and the global feature map after the max-pooling layer from the feature extraction module. Then, this combined feature map is presented to the conv mlp layers to obtain the segmentation result.

c) Classification module: It is designed to provide the target label. The global feature map from the feature extraction module is directly input to the fully connected layers to provide the classification label.

The segmentation module presents two key challenges: 1) local information mining and 2) loss function calculation. The first challenge involves extracting local information without compromising speed. The CRS module solves this issue by calculating the distance from each point in the rough segmentation block to the clicked point, thereby extracting the necessary local information.

The second challenge involves loss calculation. PointNet uses softmax cross-entropy loss (CE loss), which is inadequate for handling class imbalance, that is, situations where there are few object points and most of the points belong to the background. To address this problem, we have designed a mixed loss for segmentation, which comprises both a cross-entropy-based function and an intersection-over-union (IoU)-based function. IoU is a commonly used metric for measuring segmentation performance and can evaluate the completeness of the predicted object without being affected by the background. While IoU loss [42] has been used in the image domain, it cannot be directly converted into the point clouds domain.

The loss function of CRSNet is a combination of classification and mixed loss, which is defined as follows:

$$L_{\text{CRS}} = \gamma L_{\text{cls}} + L_{\text{mix}} \tag{5}$$

$$L_{\text{mix}} = \alpha L_{\text{IoU}} + \beta L_{\text{CE}} \tag{6}$$

$$\text{IoU} = p_s l_s / \{p_s + l_s - p_s l_s\} \tag{7}$$

$$L_{\text{IoU}} = -\log \text{IoU} \tag{8}$$

$$L_{\text{CE}} = -(1 - l_s) \log q_s - l_s \log p_s \tag{9}$$

$$L_{\text{cls}} = -l_c \log p_c \tag{10}$$

where α, β, and γ represent the balancing parameters. p_s represents the probability confidence of the points over the object, while l_s represents the segmentation label of the points. q_s represents the probability confidence of the points over the background, and l_c represents the classification label. p_c represents the probability confidence over classes.

C. Simulation Strategies

Popular public point cloud datasets are designed for common tasks such as object detection and semantic segmentation, but they lack interactive information. To obtain this information conveniently, we use a computer to randomly select points as clicks during training and testing. "Positive clicks" refer to points that belong to the target object, whereas "negative clicks" refer to points that belong to the background.

IV. Results and Discussion

To evaluate the proposed method, we conducted several experiments. We first introduce the datasets and implementation details and then evaluate our method's performance within the same datasets, as well as its generalization performance, contributions of the CRS module, and mixed loss. Finally, we analyze its robustness and other performance metrics.

A. Dataset Description and Implementation

The proposed method is evaluated using several popular point cloud datasets, including Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) [36], Apolloscape [44], Roadmarking [45], Scannet [46], and SemanticKITTI [50]. KITTI, Scannet, and SemanticKITTI are commonly used datasets in the point cloud domain, while Apolloscape's 3-D Lidar object detection and tracking dataset is similar to the KITTI detection dataset, consisting of LiDAR scanned point clouds with high-quality annotations. The Roadmarking dataset includes different road markings, similar to 2-D point clouds with intensity information. To fully test the algorithm's performance, the KITTI dataset is augmented to obtain three additional datasets: KITTI_person, KITTI_car, and KITTI_mixed. As the performance evaluation is at the point level, a simple cross-product method is used to obtain the ground-truth label of each point from the bounding box label. The transformation result is visualized in Fig. 3. Additionally, the ground-truth label is used to automatically generate clicks, and only the part of the dataset with the annotated label is used, split into 50%, 30%, and 20% subsets for training, validation, and testing, respectively. Moreover, only the geometry information (x, y, z) is used in the Scannet dataset.

KITTI_Person, KITTI_Car, and KITTI_Mixed: These three new datasets are generated from the original KITTI 3-D object detection dataset. These new datasets feature much more crowded scenes and demonstrate nonideal conditions with disturbances (other objects) and an unbalanced number

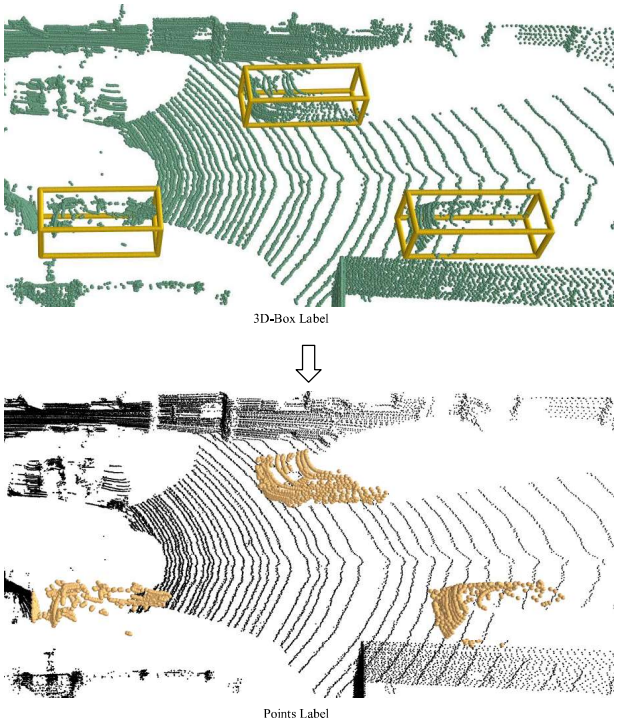

Fig. 3. Transformation from the 3-D bounding box label to the points label. The top part shows the 3-D bounding boxes on the scene. The bottom part shows the point labels on the scene. Labels are annotated in yellow.

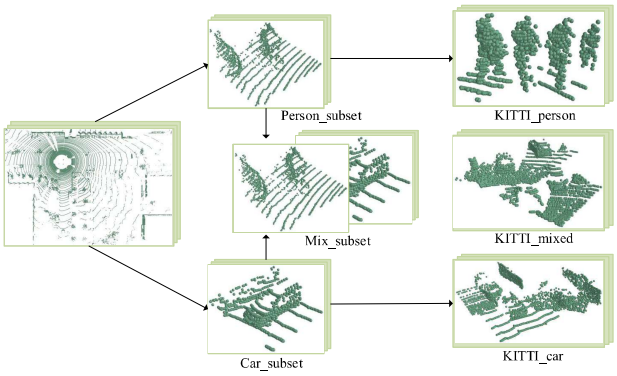

Fig. 4. Process of generating the KITTI_car, KITTI_person, and KITTI_mixed datasets from the original KITTI dataset.

of points in different categories, making the task of extracting the target object more challenging. The specific process of generating these datasets is shown in Fig. 4. First, we extracted people (including pedestrians, sitting persons, and cyclists) and cars (including cars, vans, and trucks) from each scene to form person and car subsets, respectively. Next, we stitched together three instances from the person subset to generate a new block. Using this approach, we obtained a series of blocks to form the KITTI_person dataset (7038 samples). Similarly, each sample in the KITTI_car dataset (33 672 samples) was generated by combining three instances from the car subset. Finally, we mixed the person subset and car subset to form a mixed subset, and each sample in the KITTI_mixed dataset (40 713 samples) consists of three instances from the mixed subset. These three new datasets effectively simulate crowded outdoor environments.

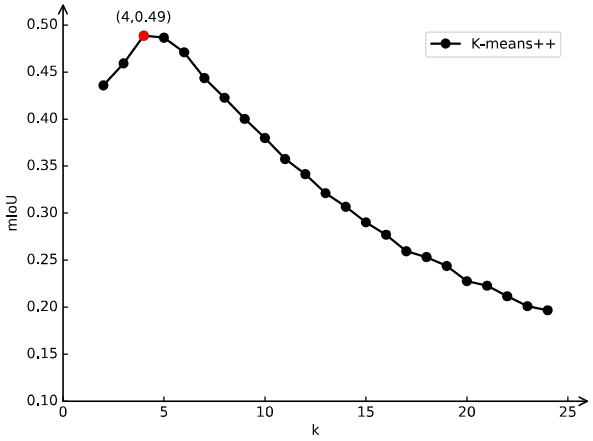

Fig. 5. Effects of different values of k on the final performance of K-means++. The red points represent the best k value and its corresponding mIoU result.

Our method produces both segmentation and classification labels, and we evaluate its performance using different metrics. For classification, we use the overall accuracy (OA) as the standard metric. For segmentation, we use the mean IoU (mIoU) as well as a popular 2-D domain metric called number of clicks (NOCs)@q% [1], [2], [3]. This metric measures the average NOCs required to reach q% intersection over union (IoU) between the predicted and ground-truth masks on objects. A lower NOC value indicates better performance. The balancing parameters of L_{CRS}: γ, α, and β [see (5), (6)] are set to 0.1, 0.1, and 1, respectively.

In addition, we conducted the proposed method on a computer with an Intel Xeon E5-2678 v3 2.50-GHz processor, 64 GB of memory, and the Ubuntu 18.04 operating system. We trained our model on a single NVIDIA GeForce 2080 Ti. The visualization of the results was achieved using MAYAVI [43].

B. Segmentation Performance Within Same Datasets

First, we trained and tested our method on the same datasets. As there is currently no typical DL-based interactive segmentation method in the point clouds domain, we compared our method with state-of-the-art methods: Mix3D [51] and 2DPASS [52] in the Scannet and SemanticKITTI 3-D semantic segmentation benchmarks. Additionally, we compared our method with traditional methods, including k-means++ and mean-shift clustering. To demonstrate the superiority of our method, we used only one click generated by a computer randomly. Our proposed method uses two backbones: PointNet and PointNeXt [24]. PointNeXt is one pioneer work in recent years that has a quick running speed and good performance in point cloud processing.

For k-means++, the value of k has a significant impact on its performance. To determine the optimal k, we conducted experiments with different k values on the KITTI dataset, and the results are presented in Fig. 5. The best-performing k value was found to be 4, which was used in subsequent experiments.

Table I presents the segmentation results on different datasets, where our method achieves higher mIoU scores than other methods on each dataset. For instance,

TABLE I

RESULTS OF DIFFERENT METHODS ON DIFFERENT DATASETS. METRIC IS MIOU (%)

Method	KITTI	KITTI_person	KITTI_car	KITTI_mixed	Apolloscape	Roadmarking	Scannet	SemanticKITTI
K-means++	48.86	33.07	37.61	36.35	10.07	20.22	1.02	3.04
Mean-shift	44.09	70.87	44.15	48.93	9.13	40.83	0.93	1.40
Mix3D [51]	-	-	-	-	-	-	73.60	59.90
2DPASS [52]	-	-	-	-	-	-	-	72.90
CRSNet(PointNet)	**94.34**	89.28	92.33	90.86	**89.74**	70.45	76.96	**95.57**
CRSNet(PointNeXt)	93.95	**91.67**	**92.7**	**91.45**	86.44	**81.32**	**81.57**	94.44

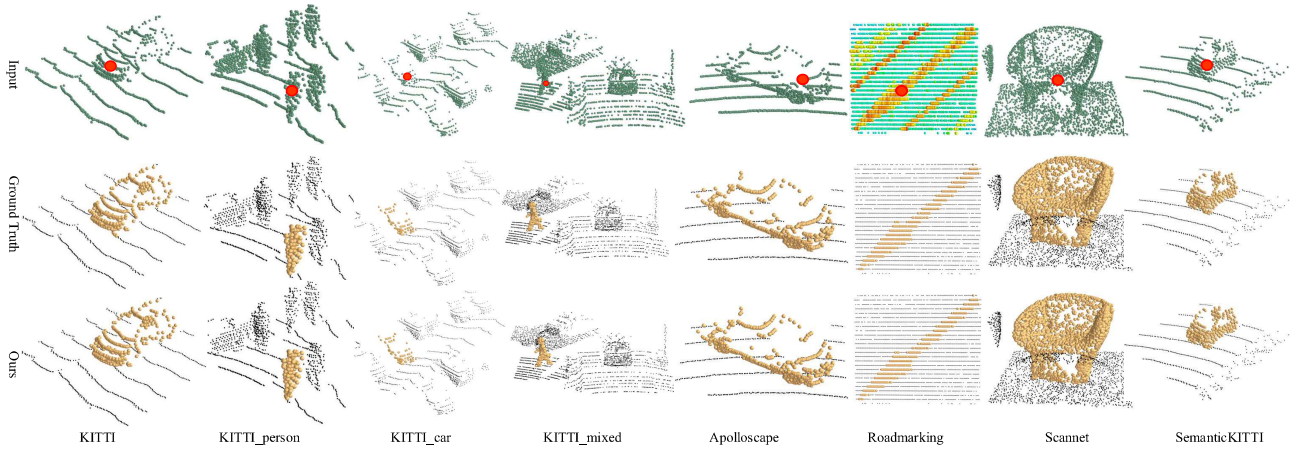

Fig. 6. Examples of segmentation results obtained by CRSNet(PointNeXt) on different datasets. The yellow points indicate the predicted object, whereas the red dot represents the click.

CRSNet(PointNeXt) achieves 81.57% and 94.44% in Scannet and SemanticKITTI datasets, respectively, which is significantly higher than that of Mix3D and 2DPASS. Therefore, our method demonstrates a more powerful segmentation ability for point clouds. This is because our method focuses only on extracting the objects of interest from the background, as required by users, while other semantic segmentation methods may not fully understand the user's intention. Moreover, compared with traditional methods, the proposed method performs better as it can extract high-level features from the point clouds and comprehend the shape of different objects, which is different from traditional methods that rely mainly on low-level features like density, distance, and connectivity, which can be easily influenced by environmental factors such as occlusion and density of points.

Fig. 6 shows the visualization of CRSNet(PointNeXt) segmentation results on different datasets, demonstrating that our method can deal with various environments, including outdoor traffic roads, indoor houses, crowds, and plain road markings.

C. Generalization Experiments

In practical applications, it is crucial for a trained model to adapt to unseen datasets or objects. To evaluate the generalization ability of our method, we tested it on datasets that are different from the training dataset. Specifically, we trained our method using the KITTI dataset and tested it on the Apolloscape dataset and Scannet dataset, respectively. The Apolloscape dataset is an outdoor environment dataset similar to KITTI, while the Scannet dataset is an indoor environment dataset that is quite different from the KITTI dataset. Through these experiments, we aimed to verify the proposed method's performance on unseen datasets. As it typically takes around eight clicks to manually draw a bounding box and precisely segment the interesting objects, we set the threshold for the maximum NOCs on each object to 10.

TABLE II

RESULTS OF GENERALIZATION EXPERIMENT ONE

(trained on KITTI)	Apolloscape			Scannet		
NOC @ k % IoU	80%	85%	90%	80%	85%	90%
CRSNet(PointNet)	**2.21**	**2.69**	**3.49**	**7.74**	**8.44**	**9.05**
CRSNet(PointNeXt)	2.46	3.10	4.15	8.63	9.07	9.41

Table II shows that even though our model is trained on KITTI only, it still achieves good segmentation performance on the Apolloscape dataset. For instance, CRSNet(PointNet) only requires 2.21, 2.69, and 3.49 clicks to reach 80%, 85%, and 90% IoU, respectively. However, for Scannet, the trained model has relatively poor performance. On average, 7.74, 8.44, and 9.05 clicks are needed to reach 80%, 85%, and 90% IoU, respectively. One reason may lie in that the indoor environment presents greater complexity compared to the outdoors. First, the spatial relationships between indoor objects are more intricate. Objects can be hung on walls, placed on other objects like tables, sofas, or beds, and sometimes they can even be concealed underneath other items, like chairs under tables. Second, indoor objects vary significantly in size, with items such as beds being much larger than vases on tables. As a result, it becomes more challenging for the network to accurately distinguish the target object from the background. Another reason may be that Scannet is captured by a depth camera whose spatial structure accuracy is significantly lower than that of Lidar. As a consequence, the texture information (excluding color) on the surface of the point cloud derived from depth cameras is considerably less abundant compared

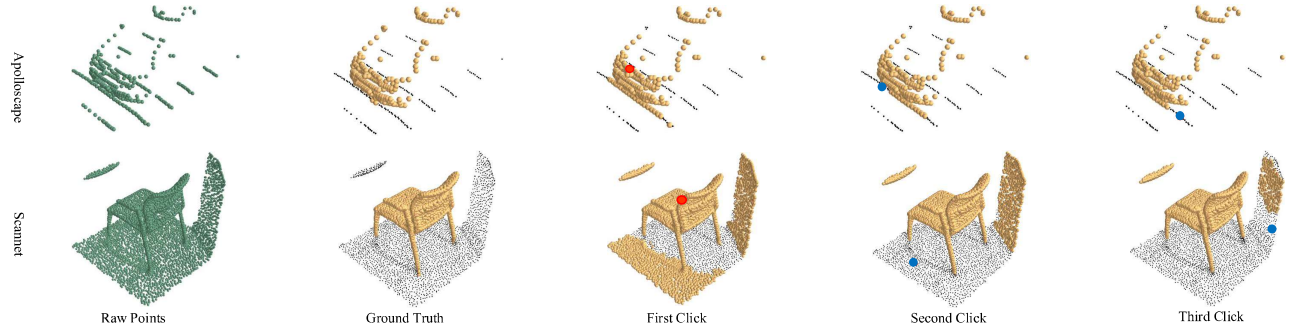

Fig. 7. Performance of CRSNet(PointNeXt) on unseen datasets. The red dot indicates the positive click, the blue dot indicates the negative click, the yellow points represent the predicted object, and the black points represent the background.

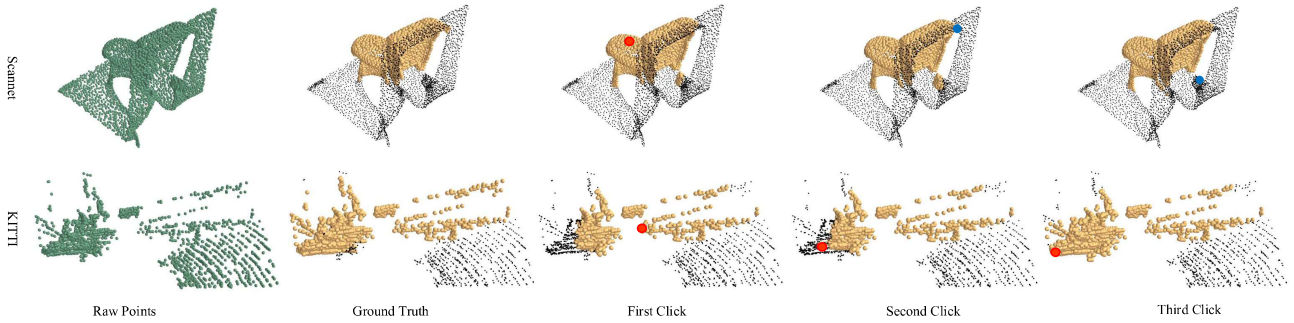

Fig. 8. Performance of CRSNet(PointNeXt) on unseen objects of different datasets. The red dot indicates the positive click, the blue dot indicates the negative click, the yellow points represent the predicted object, and the black points represent the background.

to Lidar data. This disparity in texture information may also increase the difficulty of network segmentation. Nonetheless, our proposed method has demonstrated good generalization ability to similar unseen datasets. Examples of interactive segmentation results on unseen datasets are shown in Fig. 7. Through constant clicks by the user, the segmentation results gradually approach the ground truth.

To verify the performance of our method on unseen objects, we split the dataset into two parts based on object classes. One part was used for training and the other part for testing. This ensured that all objects in the test data were unseen during training. We conducted these experiments on two different datasets: Scannet and KITTI. Scannet had 40 different object classes, but the benchmark task only categorized them into 21 classes. We trained the model on the first eight classes (excluding wall and floor) and tested it on the remaining classes. We also tested our method on the same classes as a comparison. By comparing the results of these two experiments, we could verify the performance of our proposed method on unseen objects. Additionally, we designed similar experiments on the KITTI dataset, which is an outdoor environment dataset. Specifically, we trained the model on pedestrian and car samples and tested it on the same classes as well as on other classes of objects.

Table III shows the results of testing our method on the Scannet dataset. We can observe that our method [CRSNet(PointNeXt)] requires an average of three more clicks to achieve the same IoU for unseen objects compared to seen objects. Specifically, it needs an average of 3.52, 4.56, and 6.02 clicks to reach 80%, 85%, and 90% IoU, respectively, for seen objects. In contrast, for unseen objects, the proposed method needs an average of 6.35, 7.28, and 8.27 clicks to reach the same IoU levels. The upper part of Fig. 8 shows an example of the proposed method successfully segmenting an unseen object in the Scannet dataset with just three clicks.

TABLE III

RESULTS OF GENERALIZATION EXPERIMENT TWO ON SCANNET

Scannet	Seen			Unseen		
NOC @ k % IoU	80%	85%	90%	80%	85%	90%
CRSNet(PointNet)	4.83	5.99	7.31	7.19	7.99	8.82
CRSNet(PointNeXt)	**3.52**	**4.56**	**6.02**	**6.35**	**7.28**	**8.27**

TABLE IV

RESULTS OF GENERALIZATION EXPERIMENT TWO ON KITTI

KITTI	Seen			Unseen		
NOC @ k % IoU	80%	85%	90%	80%	85%	90%
CRSNet(PointNet)	**1.09**	**1.16**	**1.31**	**1.77**	**2.10**	**2.89**
CRSNet(PointNeXt)	1.23	1.38	1.75	2.10	2.48	3.19

Table IV presents the results of testing the proposed method on the KITTI dataset. The results indicate that the proposed method [CRSNet(PointNet)] requires only slightly more clicks to achieve the same IoU for unseen objects compared to seen objects. Specifically, it requires an average of 1.09, 1.16, and 1.31 clicks to achieve 80%, 85%, and 90% IoU, respectively, for seen objects, and an average of 1.77, 2.10, and 2.89 clicks to achieve the same IoU for unseen objects. The lower part of Fig. 8 shows an example where the proposed method successfully segments unseen objects in the KITTI dataset through several clicks. The generalization performance of the proposed method in KITTI is better than that in Scannet, possibly because Scannet lacks RGB information, making it difficult to distinguish interesting objects from other things, and has more challenging object classes than KITTI. Overall, the results demonstrate that the proposed method can segment unseen objects.

TABLE V
RESULTS OF ABLATION EXPERIMENTS ON FOUR DATASETS. THE METRIC IS mIoU (%)

Method	KITTI	KITTI_person	KITTI_car	KITTI_mixed
PointNet	93.54	0.00	0.00	0.00
PointNet+CRS	**94.49**	89.00	0.00	0.00
PointNet+mixed loss	93.65	0.00	0.00	0.00
Ours(PointNet+CRS+mixed loss)	94.34	**89.28**	**92.33**	**90.86**

In summary, the proposed method can segment unseen datasets to a certain extent. If the test dataset is similar to the training dataset, such as being captured in a similar environment and by similar devices, the proposed method may perform well. However, if the test dataset is significantly different from the training dataset, the proposed method may have a relatively poorer performance. With regard to unseen objects, our method can segment them.

D. Ablation Experiments

The CRS module and the mixed loss function are two crucial components of the proposed method. The CRS module is designed to obtain point clouds with interactive information, while the mixed loss function guides the model's training. To evaluate their influence, we conducted ablation experiments on different datasets. "PointNet + CRS" and "PointNet + mixed loss" denote PointNet with only the CRS module and mixed loss, respectively. The compared results are shown in Table V, where each object only received one positive click.

As shown in Table V, all methods perform well on the KITTI dataset, possibly because objects in KITTI are usually not adjacent. On the KITTI_person dataset, "PointNet + CRS" achieves an mIoU of 89.00%, while PointNet fails to complete the task. Without the CRS module, PointNet alone cannot focus on the target object and locate it for segmentation. Our proposed method achieves 92.33% mIoU on KITTI_car and 90.86% mIoU on KITTI_mixed, while "PointNet + CRS" fails in these two datasets. Without the mixed loss, "PointNet + CRS" classifies all points as background, as the background points make up most of the entire scene. Comparing "PointNet + mixed loss" with our method, our method performs well in KITTI_person, KITTI_car, and KITTI_mixed, while "PointNet + mixed loss" fails. This is because "PointNet + mixed loss" cannot identify which object to segment without the CRS module.

The above results demonstrate the contribution of the CRS module and mixed loss. Our proposed method, with the CRS module, can resist interference from other objects, especially in datasets like KITTI_person, KITTI_car, and KITTI_mixed, where the interference is very severe. Additionally, the CRS module not only provides the position of the object, but also gives local information, which helps the network to capture the local information around the click center. This, in turn, improves the performance of the network.

E. Influence of the Number of Training Samples

In this section, we evaluate the influence of the number of training samples on the performance of our proposed model. As a DL method, our model requires a significant number of training samples. However, in practice, labeling these training samples is time-consuming and labor-intensive. Therefore, it is essential to ensure that the model is robust to the number of training samples. We conducted experiments where we gradually reduced the number of training samples from 100% to 2% and analyzed the segmentation performance on the same test dataset with one positive click. The results of these experiments can be found in Figs. 9 and 10.

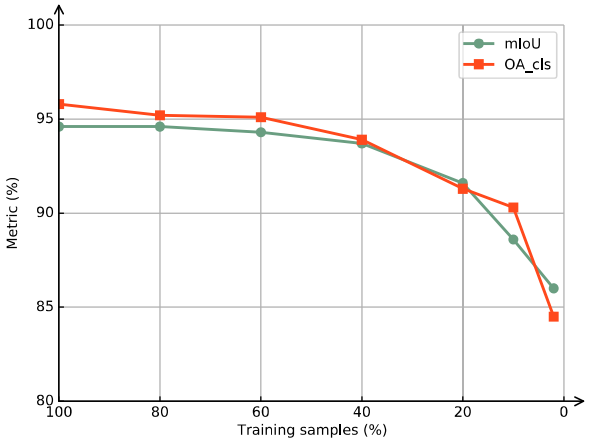

Fig. 9. Performance of the proposed method with a different number of training samples.

From Fig. 9, we can see that our method is robust to changes in the size of the training dataset. Specifically, the performance of the network decreases only slightly as the number of training samples is reduced. For example, when the number of samples decreases from 100% to 20%, the performance only drops by 3% in mIoU. Furthermore, even when the test dataset size is larger than the training dataset size, our method still achieves satisfactory segmentation results. As demonstrated in Fig. 10, even with only 2% of training samples, the target object can still be fully segmented. For instance, in the upper example, the car can be successfully segmented with one click.

F. Influence of Parameters

The balancing parameters of L_{CRS}: γ, α, and β [see (5), (6)] play a crucial role in the segmentation and classification performance. Therefore, conducting experiments to investigate the influence of these parameters on performance is necessary.

First, we designed experiments on CRSNet(PointNet) using the KITTI_mixed dataset to examine the effects of different α and β values on L_{mix} [see (6)]. Specifically, we set γ to 0 and β to 1, then continuously varied the value of α from 0.1 to 1. Additionally, we used only one positive click to point out the interesting object. As shown by the green line in Fig. 11, we observed that as α increases from 0.1 to 1, the segmentation performance remains stable, hovering around 91%.

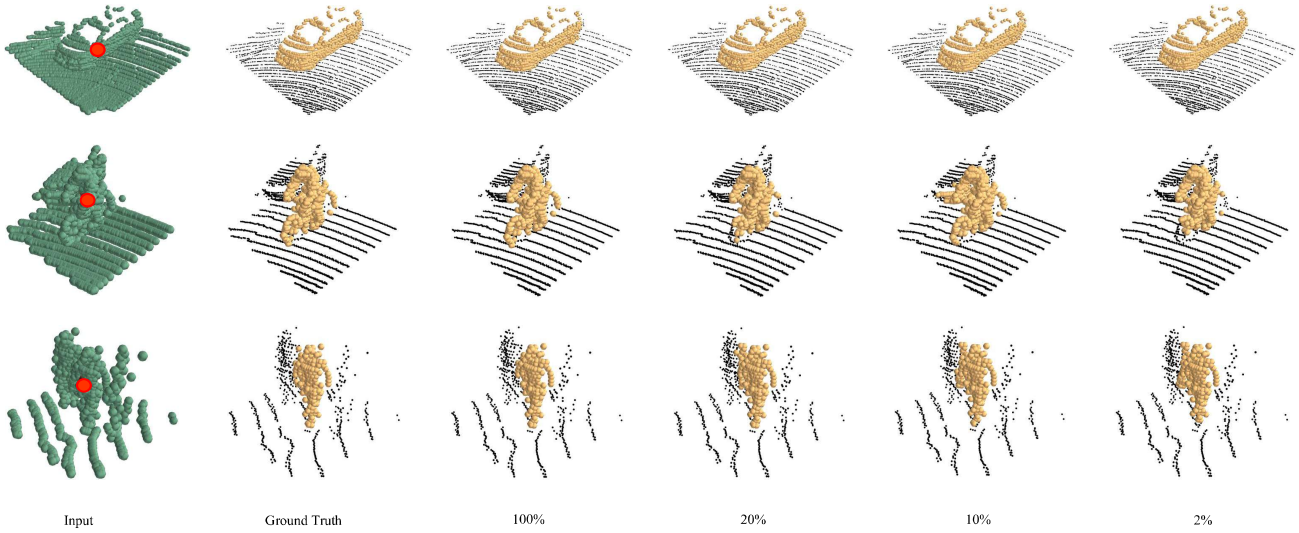

Fig. 10. Test results with a different number of samples on KITTI. Red dots indicate clicks made by the user, whereas yellow points represent points belonging to interesting objects.

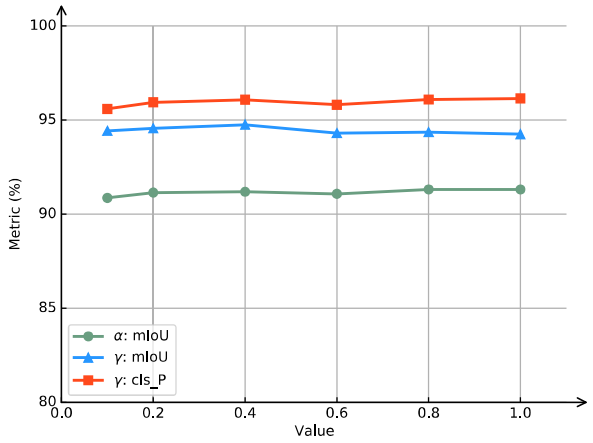

Fig. 11. Performances of CRSNet(PointNet) with different parameter values. The green line represents the segmentation performance with varying α values. The blue line and orange line display the segmentation performance and classification performance, respectively, with different γ values.

Hence, it can be concluded that different values of the parameters α and β have minimal impact on the segmentation results. The reason behind this is that both the IoU loss and CE loss are designed to effectively segment objects, thus changes in these parameters have little effect on the segmentation accuracy.

While classification is not the primary aspect of the interactive segmentation task, it can facilitate automatically obtaining the semantic label of the segmented object. To test the effect of the classification parameter γ on segmentation and classification, we conducted further experiments on CRSNet(PointNet) using the KITTI dataset, where one positive click was used to point out the interesting object. Specifically, we set β to 1, α to 0.1, and continuously varied the value of γ. The experimental results are shown in Fig. 11, with the blue line representing segmentation performance and the orange line representing classification performance. As observed, increasing γ had minimal impact on both segmentation and classification performances. This is likely due to the point cloud feature information obtained by the feature extraction module being rich enough to simultaneously achieve the segmentation and classification functions.

V. Conclusion

In this study, we presented a novel framework for point cloud interactive segmentation, called CRSNet, which comprises two main components: the CRS module and the feature extraction network. We evaluated the segmentation performance, generalization abilities across different datasets and object classes, the contributions of the CRS module, and mixed loss, robustness, and other metrics. Our proposed method offers several advantages.

1) The interactive process is simple, as users need only click on the target object to obtain a segmented result.
2) Users can constantly click on the segmentation result until a satisfactory outcome is achieved.
3) CRSNet can adapt to unseen object classes and unseen datasets. Besides, it is robust to variations in the number of training samples, making it suitable for practical applications such as annotation.
4) The analysis module backbone can be easily replaced with different DL frameworks, allowing for easy updates to our method.

Our comparative experiments demonstrate that CRSNet outperforms state-of-the-art methods in semantic segmentation. The generalization experiments further demonstrate that our method can accurately segment unseen datasets and objects. The ablation experiments show that the CRS module and mixed loss both contribute significantly to the final results. Additionally, the robustness experiments demonstrate that our method is suitable for practical applications. Therefore, we can conclude that our proposed method can accurately, efficiently, and robustly achieve interactive segmentation for point clouds.

In the future, we plan to refine our method in the following aspects: first, we aim to reduce the NOCs required for fully segmenting objects in indoor environments. Second, we will work toward improving the performance of our method in generalization experiments.

REFERENCES

[1] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. CVPR*, 2016, pp. 373–381.

[2] Z. Li, Q. Chen, and V. Koltun, "Interactive image segmentation with latent diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 577–585.

[3] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, and S.-P. Lu, "Interactive image segmentation with first click attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13336–13345.

[4] M. Kassis and J. El-Sana, "Scribble based interactive page layout segmentation using Gabor filter," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 13–18.

[5] T. Yajima, Y. Kanamori, Y. Endo, and J. Mitani, "Interactive edge-aware segmentation of character illustrations for articulated 2D animations," in *Proc. Nicograph Int. (NicoInt)*, Jun. 2018, pp. 1–8.

[6] H. Yu, Y. Zhou, H. Qian, M. Xian, and S. Wang, "Loosecut: Interactive image segmentation with loosely bounded boxes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3335–3339.

[7] A. S. Boroujerdi, M. Khanian, and M. Breuß, "Deep interactive region segmentation and captioning," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Dec. 2017, pp. 103–110.

[8] S. Wu, M. Nakao, and T. Matsuda, "SuperCut: Superpixel based foreground extraction with loose bounding boxes in one cutting," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1803–1807, Dec. 2017.

[9] A. S. A. Khaizi, R. A. M. Rosidi, H.-S. Gan, and K. A. Sayuti, "A mini review on the design of interactive tool for medical image segmentation," in *Proc. Int. Conf. Eng. Technol. Technopreneurship (ICET)*, Sep. 2017, pp. 1–5.

[10] H. Jinbo, T. Kitrungrotsaku, Y. Iwamoto, L. Lin, H. Hu, and Y.-W. Chen, "Development of an interactive semantic medical image segmentation system," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Oct. 2020, pp. 678–681.

[11] Y. Heo, Y. J. Koh, and C.-S. Kim, "Inter-image affinity based interactive video object segmentation," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 384–386.

[12] W.-H. Li et al., "An interactive segmentation method of LiDAR data," *J. Comput.*, vol. 8, no. 3, pp. 811–817, Mar. 2013.

[13] H. Luo, Q. Zheng, C. Wang, and W. Guo, "Boundary-aware and semi-automatic segmentation of 3-D object in point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 5, pp. 910–914, May 2021.

[14] K. Liu and J. Boehm, "A new framework for interactive segmentation of point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 5, pp. 357–362, Jun. 2014.

[15] H. Luo et al., "Boundary-aware graph Markov neural network for semiautomated object segmentation from point clouds," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 104, Dec. 2021, Art. no. 102564.

[16] Z. Tang and Z. Miao, "Interactive foreground extraction for photo and video editing," in *Proc. 9th Int. Conf. Signal Process.*, Oct. 2008, pp. 1483–1486.

[17] X. Wu and Y. Wang, "Interactive foreground/background segmentation based on graph cut," in *Proc. Congr. Image Signal Process.*, 2008, pp. 692–696.

[18] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[19] 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. 77–85.

[20] 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*.

[21] 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, Oct. 2019.

[22] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. NIPS*, 2018, pp. 828–838.

[23] Q. Hu et al., "Learning semantic segmentation of large-scale point clouds with random sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Nov. 2022.

[24] G. Qian et al., "PointNeXt: Revisiting PointNet++ with improved training and scaling strategies," in *Proc. NIPS*, 2022, vol. 35, pp. 23192–23204.

[25] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proc. Annual Conf. Comp. Graph. Interac. Tech.*, 1995, pp. 191–198.

[26] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 105–112.

[27] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[28] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, Aug. 2004.

[29] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[30] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *Proc. ECCV*, 2008, pp. 264–275.

[31] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng, "Regional interactive image segmentation networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2746–2754.

[32] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via back-propagating refinement scheme," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5292–5301.

[33] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.

[34] D. Arthur and S. Vassilvitskii, "Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 153–164.

[35] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[36] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[37] 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.

[38] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4416–4425.

[39] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[40] L. Weilong, "Interactive clothing image segmentation based on super-pixels and graph cuts," in *Proc. Int. Conf. Comput. Sci. Appl.*, Dec. 2013, pp. 659–662.

[41] 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.

[42] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.

[43] P. Ramachandran and G. Varoquaux, "Mayavi: 3D visualization of scientific data," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 40–51, Mar. 2011.

[44] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 6120–6127.

[45] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, and A. Habib, "A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 178–192, Jan. 2019.

[46] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[47] X. Xu and G. H. Lee, "Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13706–13715.

[48] J. Wei, G. Lin, K.-H. Yap, T.-Y. Hung, and L. Xie, "Multi-path region mining for weakly supervised 3D semantic segmentation on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4383–4392.

[49] C.-K. Yang, J.-J. Wu, K.-S. Chen, Y.-Y. Chuang, and Y.-Y. Lin, "An MIL-derived transformer for weakly supervised point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11820–11829.

[50] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.

[51] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann, "Mix3D: Out-of-context data augmentation for 3D scenes," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 116–125.

[52] X. Yan et al., "2DPASS: 2D priors assisted semantic segmentation on LiDAR point clouds," in *Proc. ECCV*, 2022, pp. 677–695.

Wentao Sun received the M.Sc. degree in electronics and communication engineering from the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China, in July 2022.

He is currently a Teaching Assistant with the Institute of Artificial Intelligence and the Department of Computer Science and Engineering, School of Mechanical and Electrical Engineering, Shaoxing University, Shaoxing, China. His research interests include 3-D point cloud semantic segmentation, 3-D point cloud instance segmentation, interactive segmentation, and computer vision.

Zhipeng Luo (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China, in July 2020.

From 2021 to 2022, he was a Post-Doctoral Researcher with Hong Kong Polytechnic University, Hong Kong. He is currently an Associate Professor with the Department of Computer Science, Minnan Normal University, Zhangzhou, China. His research interests include autonomous driving, mobile laser scanning, intelligent processing of point clouds, 3-D computer vision, and machine learning.

Yiping Chen (Senior Member, IEEE) received the Ph.D. degree in information and communications engineering from the National University of Defense Technology, Changsha, China, in 2011.

She is currently an Associate Professor with the School of Geospatial Engineering and Science, Sun Yat-sen University, Zhuhai, China. She was an Assistant Researcher with the Chinese University of Hong Kong, Hong Kong, from 2007 to 2011 followed by a Research Associate Professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, China, from 2011 to 2022. She has published more than 80 research papers in referred journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and Flagship Conferences, including CVPR, IGARSS, and ISPRS. Her research interests include remote-sensing image processing, mobile laser scanning data analysis, 3-D point cloud analytics, computer vision, and autonomous driving.

Dr. Chen was a recipient of the 2020 Best Reviewer of IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She is an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Huxiong Li received the M.Sc. degree in software engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in pattern recognition and intelligent system from the Northwest University of Technology, Xi'an, China, in 2012.

He is currently a Professor with the Department of Computer Science and Engineering, School of Mechanical and Electrical Engineering, and an Executive Director of the Institute of Artificial Intelligence, Shaoxing University, Shaoxing, China. His research interests include network control, complex networks, knowledge graphs, machine learning, 3-D computer vision, and cloud computing.

José Marcato Junior (Member, IEEE) received the Ph.D. degree in cartographic science from Sao Paulo State University, São Paulo, Brazil, in 2014.

He is currently a Professor with the Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande, Brazil. He has published more than 40 research papers in refereed journals and more than 70 in conferences, including *ISPRS Journal of Photogrammetry and Remote Sensing*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. His research interests include UAV photogrammetry and deep neural networks for object detection, classification, and segmentation.

Wesley Nunes Gonéalves (Member, IEEE) received the Ph.D. degree in computational physics from the University of Sao Paulo, São Paulo, Brazil, in 2013.

He is currently a Professor with the Faculty of Computer Science and Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande, Brazil. He has published more than 40 in refereed journals and more than 60 in conferences, including *Pattern Recognition*, *Pattern Recognition Letters*, *Neurocomputing*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. His research interests include computer vision, machine learning, and deep neural networks for object detection, classification, and segmentation.

Jonathan Li (Fellow, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000.

He is currently a Professor of Geomatics and Systems Design Engineering with the University of Waterloo, Waterloo, ON, Canada. He has coauthored more than 530 publications, more than 330 of which were published in refereed journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, *ISPRS Journal of Photogrammetry and Remote Sensing*, *Remote Sensing of Environment*, and *Journal of Applied Gerontology*. He has also published papers in flagship conferences in computer vision and AI, including CVPR, AAAI, and IJCAI. He has supervised more than 120 master's and Ph.D. students as well as post-doctoral fellows to completion. His main research interests include AI-based information extraction from earth observation images and LiDAR point clouds, pointgrammetry, 3-D vision, and GeoAI for digital twin cities.

Dr. Li is the Editor-in-Chief of *International Journal of Applied Earth Observation and Geoinformation* (JAG) and an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and *Canadian Journal of Remote Sensing*. He is a fellow of *Canadian Academy of Engineering* and ENGINEERING INSTITUTE OF CANADA.