Article

Instance Segmentation of LiDAR Point Clouds with Local Perception and Channel Similarity

Xinmiao Du * and Xihong Wu *

Posted Date: 7 August 2025

doi: 10.20944/preprints202508.0572.v1

Keywords: LiDAR; point clouds; instance segmentation; channel attention mechanism; inter-channel similarity; local perception

Article

Instance Segmentation of LiDAR Point Clouds with Local Perception and Channel Similarity

Xinmiao Du * and Xihong Wu *

School of Intelligence Science and Technology, Peking University, Beijing 100080, China

* Correspondence: 2001111389@stu.pku.edu.cn; wxh@cis.pku.edu.cn

Abstract

LiDAR and its point cloud data are crucial visual sensors in smart driving cars for sensing the surrounding environment and achieving high accuracy localization. Compared to semantic segmentation, point cloud instance segmentation is a more complicated task. For autonomous driving systems, precise instance segmentation results can offer a more detailed understanding of the scene. The point cloud data is sparse and irregular, and the point cloud density varies with the distance from the sensor. In this paper, a LiDAR channel-aware point segmentation network (LCPSNet) is proposed to address the above problems. Given the distance-dependent sparsity and drastic scale variation of LiDAR, we adopt a top-down FPN. High-level features are progressively upsampled and summed element-wise with the corresponding shallow layers. Beyond this standard fusion, the fused features at 1/16, 1/8, and 1/4 are resampled to a common BEV/polar grid. These aligned features are then fed to the LPM to perform cross-scale, position-dependent weighting and modulation at the same spatial locations. The Local Perception Module (LPM) uses global spatial information to guide, while preserving, attention to group (scale) differences. Position-by-position weighting and re-fusion of local features of each group on the same grid. Enhances both intra-object and tele-context while suppressing cross-instance interference. The Inter-Channel Correlation Module (ICCM) uses a ball query to define local feature regions, modeling the spatio-temporal and channel information of the LiDAR point cloud jointly within the same module. And the inter-channel similarity matrix is computed to remove redundant features and highlight valid features. Experiments on SemanticKITTI and Waymo datasets verify that the two modules effectively improve the local feature and global semantic consistency modeling capabilities, respectively. The PQ of LCPSNet on the SemanticKITTI dataset is 70.9, and the mIoU is 77.1, and the instance segmentation performance exceeds the existing mainstream methods and reaches SOTA.

Keywords: LiDAR; point clouds; instance segmentation; channel attention mechanism; inter-channel similarity; local perception

1. Introduction

With the rapid development of 3D sensing technology, the application of LiDAR has made the acquisition of point cloud data increasingly convenient and efficient [1], showing great application potential in many fields, including autonomous driving, robot navigation, building information modeling, and virtual reality [2–4]. The core processing tasks of point clouds include classification and segmentation of point clouds [5,6]. Among these challenging tasks, instance segmentation task for point clouds because it not only requires the algorithm to distinguish different categories but also to correctly distinguish each independent instance within the same category [7].

2D instance segmentation methods have been well-developed, and following the top-down paradigm, they can effectively segment instances in images. However, due to complex interactions and occlusions between 3D instances, pixel-based 2D segmentation methods cannot be directly applied to process unstructured and sparse 3D point clouds.

In recent years, alongside the rapid advancement of deep learning approaches [8–11], a series of instance segmentation networks grounded in deep learning have been put forward one after another. These methods can be broadly categorized into top-down-based and bottom-up-based [12,13]. The top-down method first identifies the region where the instance is located based on object detection methods, then filters the image features through non-maximum suppression, and refines them through mask segmentation. The bottom-up method distinguishes different instances through clustering and metric learning based on points with the same semantics in the semantic segmentation results. However, both methods still have obvious drawbacks. The limitation of the former lies in its dependence on the quality of bounding boxes, and the uneven distribution of point clouds will lead to unstable quality of bounding boxes, thus affecting the quality of instance segmentation. The latter has the problem of loss of local feature information after voxelization, and excessive post-processing steps will increase the amount of computation, while it is difficult to improve the segmentation performance in complex scenes.

To address issues such as insufficient capability in extracting local features and redundant repeated features in existing technologies, this paper innovatively proposes a LiDAR Channel-Aware Point Segmentation Network named LCPSNet. The network takes local perception and channel similarity as its core design concepts, with innovations reflected in the following three aspects.

(1) The Local Perception Module (LPM) performs location-dependent local perception and fusion of multi-scale/multi-neighborhood point cloud features on a uniform BEV/polar coordinate grid. The multi-scale local features are first spliced in the group dimension, and then a lightweight convolutional branch generates the global spatial saliency map and group-related components. Position-by-position modulation of local features at each scale is followed by fusion. This mechanism is more effective against distance sparsity and occlusion in LiDAR than conventional FPN/CBAM that only does channel weighting or in-scale fusion.

(2) The Inter-Channel Correlation Module (ICCM) is proposed, which innovatively introduces the channel attention mechanism in the clustering stage. By calculating the attention weight matrix between feature channels, ICCM adaptively enhances effective channel features and removes redundant channel features. Thus, the representative features of the target instances in the point cloud are extracted more effectively.

(3) Aiming at the distance-dependent sparsity and drastic scale changes of LiDAR, a position-aligned multi-scale fusion is designed based on the top-down path of FPN. It not only up-samples and sums the corresponding shallow layers element by element but also uniformly resamples the fused features of 1/16, 1/8, 1/4, and other layers to the same BEV/polar coordinate grid. The fusion features of 1/16, 1/8, 1/4, and other layers are uniformly resampled to the same BEV/polar coordinate grid, and then the LPM performs cross-scale position-dependent weighting and modulation at the same spatial location.

2. Related Work

In 3D point clouds, 3D instance segmentation can accurately tell apart distinct categories as well as separate instances within the same category, thus providing key technical support to fields including autonomous driving and robot navigation. Research on 3D instance segmentation-related technologies promotes the innovation and performance improvement of point cloud processing algorithms. Deep learning-based instance segmentation network methods are mainly divided into top-down-based and bottom-up-based, which is shown in Figure 1.

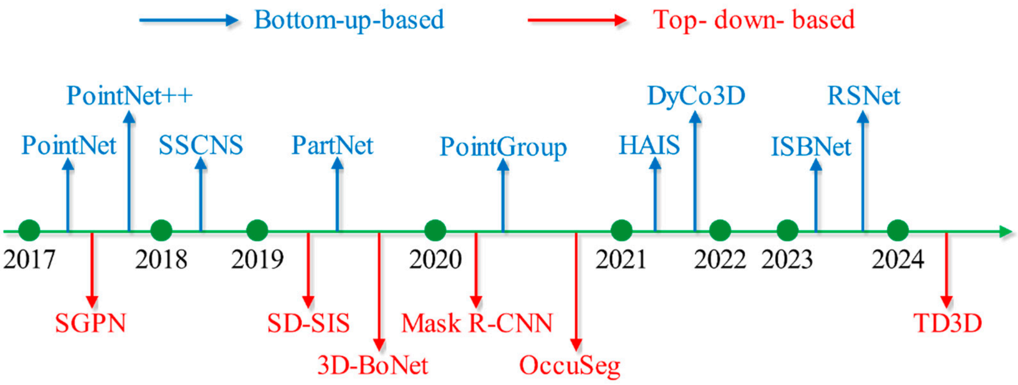


Figure 1. Deep learning-based 3D instance segmentation methods.

2.1. Top-Down-Based Methods

In the top-down based methods, the system first generates potential object proposals in 3D space, that is, predicts the bounding boxes where each entity may exist. As one of the early 3D instance segmentation methods, SGPN segments point clouds with the help of point group networks and completes instance clustering by learning the similarity between points, laying the foundation for the development of subsequent 3D instance segmentation technologies [14]. However, the SGPN method exhibits limited performance when handling high-complexity data. In contrast, the 3D SIS method, which applies the Mask R-CNN architecture to instance segmentation [15,16], enhances instance segmentation precision by extracting image features from both RGB images and 3D point clouds. But the detection network and segmentation network of 3D-SIS do not share networks and weights, and the data processing process is relatively complex.

3D-BoNet emphasizes the significance of network depth in capturing complex features of 3D data. It performs instance segmentation by directly regressing 3D bounding boxes and predicting point-level masks for each instance, thereby reducing reliance on post-processing steps [17]. However, as point clouds are distributed unevenly, bounding box quality tends to be unstable, which affects the quality of instance segmentation. OccuSeg [18] is mainly designed to address the sparsity of point cloud data. Nevertheless, this method still faces many technical challenges in determining the position of bounding boxes, fitting bounding boxes to cover target entities, and handling incompleteness in point cloud data. These challenges often lead to inaccurate positioning of bounding boxes, which in turn affects the quality of instance segmentation. TD3D conducts fully convolutional end-to-end training in a data-driven manner without relying on information of detected objects, reducing the burden of manual parameter tuning [19].

However, all top-down methods depend on the quality of bounding boxes. Given the uneven distribution of point clouds, the quality of these bounding boxes tends to be unstable, thereby impairing the performance of instance segmentation.

2.2. Bottom-Up-Based Methods

These methods are based on the results of point cloud semantic segmentation and complete segmentation by aggregating points with the same semantics into individual instances. PointNet++, built on the pioneering PointNet, addresses issues such as sparsity, disorder, and permutation invariance in point clouds by independently extracting features from all points and aggregating global features [20,21]. It also introduces adaptive sampling and points cloud partitioning strategies, extracting features from each region using PointNet before aggregating them. PartNet uses PointNet++ as the backbone network to predict semantic labels for each point and incorporates a segment detection network to achieve point cloud instance segmentation [22]. SSTNet effectively handles dynamic scenes through innovative methods, enhancing the ability to extract spatiotemporal features, especially excelling in processing moving objects [23]. PointGroup processes instance segmentation tasks in point clouds using dual-set point grouping, which has advantages in speed and significantly improves runtime performance through simplified strategies [24]. However, the success of this strategy largely depends on accurate semantic predictions, and even minor errors can lead to significant deviations in instance segmentation results.

Some new technologies such as HAIS have achieved high accuracy in 3D instance segmentation tasks with hierarchical network design and effective feature fusion strategies. However, their complex network structure increases computational requirements and shows weak segmentation accuracy in complex scenes [25]. DyCo3D adopts dynamic convolution technology, without a predefined proposal framework, and can directly generate instance masks through a small number of simple convolution layers [26]. This design dynamically generates convolution filters by responding to the spatial distribution of data and semantic predictions, thereby enhancing the ability to distinguish different instances. However, it relies heavily on centroid offset prediction, and due to insufficient local feature extraction capabilities, it is prone to incorrect grouping in dense object scenarios. ISBNet includes an instance farthest point sampling and box-aware mechanism, which can improve point recall while enhancing features through geometric clues [27]. As a 3D point cloud segmentation approach based on a recursive slicing network, TD3D technology realizes point cloud semantic and instance-wise segmentation through slicing processing and the BiLSTM [28]. However, this method is limited by the processing capacity of BiLSTM and may encounter gradient vanishing problems in complex environments, which in turn limits its performance.

Although these novel bottom-up methods have achieved good performance, new problems have emerged. This type of method will lose some local feature information after voxelization. DyCo3D is more dependent on centroid offset prediction. Some methods add excessive post-processing steps to improve accuracy, such as the segment detection network introduced by PartNet, the small Transformer introduced by DyCo3D, and the box-aware mechanism of ISBNet. Through these post-processing steps, the network can extract more feature information. However, while increasing the amount of computation, these methods cannot improve the instance segmentation performance for complex scenes, resulting in more repeated or merged invalid 3D information in scale, and the segmentation efficiency and accuracy are not ideal.

3. Methods

3.1. The Overall Structure of the LiDAR Channel-Aware Point Segmentation Network

In this paper, a LiDAR Channel-Aware Point Segmentation Network (LCPSNet) is proposed. It is an end-to-end instance segmentation network composed of five main parts including the backbone network, prediction module, Local Perception Module (LPM), Inter-Channel Correlation Module (ICCM) and instance decoder. LCPSNet takes the original point cloud 3D coordinates and RGB color vectors as input. First, the point cloud is first voxelized into a raster representation and fed into the backbone network, where semantic and geometric features are extracted at scales such as 1/2, 1/4, 1/8, 1/16, and 1/32. The top-down path up-samples (up×N) the high-level features and sums them with the corresponding shallow features element by element to realize FPN multi-scale fusion. Next, the fused multi-scale features enter the LPM. Location-dependent spatial weights are generated and weighted and fused to the local features at different scales to enhance the response within the object and suppress cross-instance interference at the boundaries. These features are then passed to the Prediction Module containing the Offset Prediction Linear Layer and the Semantic Prediction Linear Layer. The instance decoder then clusters/correlates the semantic masks with the offset fields to generate instance masks. The immediately following ICCM further refines the instance and semantic results to output the final instance segmentation. The network structure of LCPSNet is shown in Figure 2. Its overall mechanism is multiscale fusion through a feature pyramid in the backbone, followed by local perceptual weighting using LPM, and then joint decoding using semantics and offsets. Finally, ICCM further refines the instance features through ball query and a three-dimensional attention mechanism.

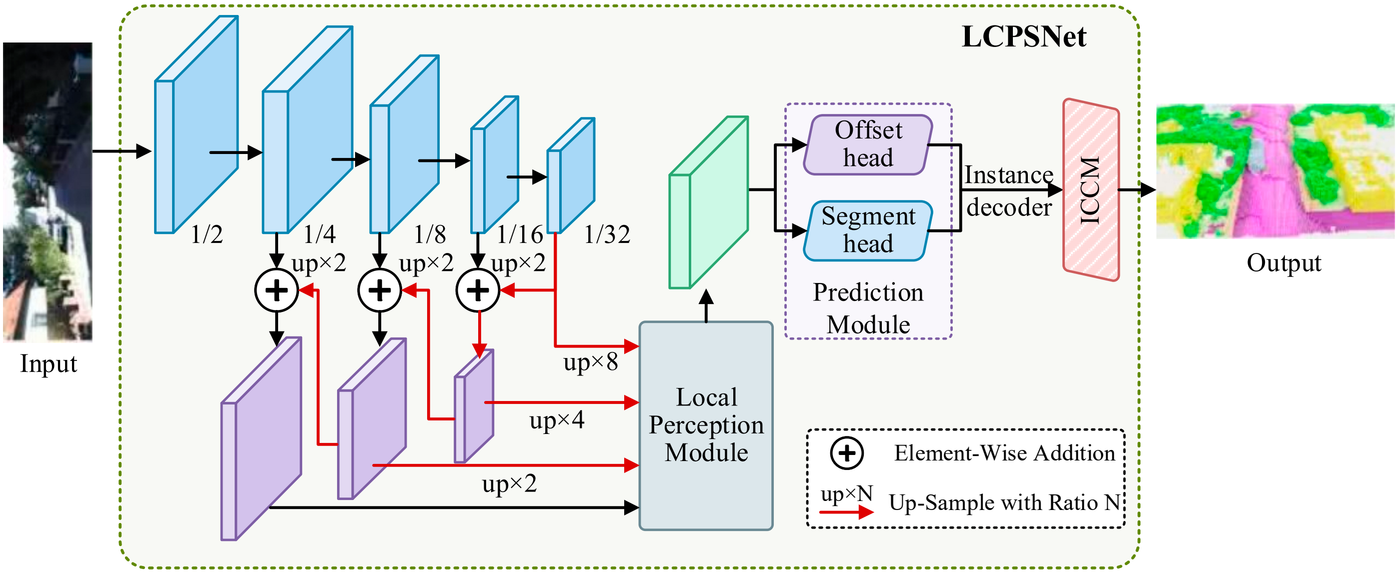


Figure 2. The network structure of LCPSNet.

The backbone in LCPSNet is a top-down multiscale fusion, as shown in Figure 2. The point cloud is first voxelized to a fixed-resolution grid, and the backbone extracts pyramidal features at steps 1/2, 1/4, 1/8, 1/16, and 1/32. Deeper features have stronger semantics and larger sensory fields and are able to cover context at longer distances and large targets, but with coarser spatial localization. Shallow features, on the other hand, retain geometric details for close range and small targets. The top-down path is upsampled by a factor of 2 at each level, starting from the coarsest 1/32, to match the resolution of the next level of features. At each level, the channel is aligned using a 1×1 convolution and then summed element-by-element with the shallow features at the same level, thus injecting the deep global semantics into the shallow fine geometry. This process is performed sequentially at 1/16, 1/8, and 1/4 to obtain a fused feature map with both context and boundary information. The up-sampling implementation can use nearest-neighbor/bilinear up-sampling followed by 3×3 convolution (range-view) or back-convolution (BEV) to mitigate aliasing and stabilize training.

In order to allow the subsequent Local Perception Module to perform location-dependent weighting on the same spatial grid, the network resamples each of the fused features at 1/16, 1/8, and 1/4 scales to a uniform target resolution. 8x magnification from 1/16, 4x magnification from 1/8, and 2x magnification from 1/4, and feeds them into the LPM along with the 1/2 features that are themselves at the target resolution. Doing so allows position-by-position alignment and modulation of responses from different receptive fields on the same polar coordinate unit. Distant or large objects rely on the contextual enhancement brought by the deep layers, while close or small objects benefit from the fine boundaries preserved by the shallow layers. This improves the separability of neighboring instances in sparse, occluded, and scale-variable LiDAR scenes.

3.2. Local Perception Module

Considering the sparse nature of LiDAR point cloud data itself, local feature information is inevitably lost during systematic processing. This lack of information cannot fully reflect the local details of the LiDAR point cloud data, which in turn may adversely affect the subsequent processing steps and reduce the accuracy of segmentation. Therefore, in this paper, the Local Perception Module (LPM) is designed to follow the backbone. This module enriches the contextual information of each point by extracting the structural information between neighboring points through local and global attention mechanism. Subsequently, enhancement of the features contained in each point is achieved by combining the local features with the global features through a dynamic fusion module. This approach better maintains the integrity of the local features when dealing with LiDAR point cloud tasks. This enhances the generalization ability of the model and the perception of multi-scale features and ultimately improves the segmentation performance of the network. The overall structure of LPM is shown in Figure 3.

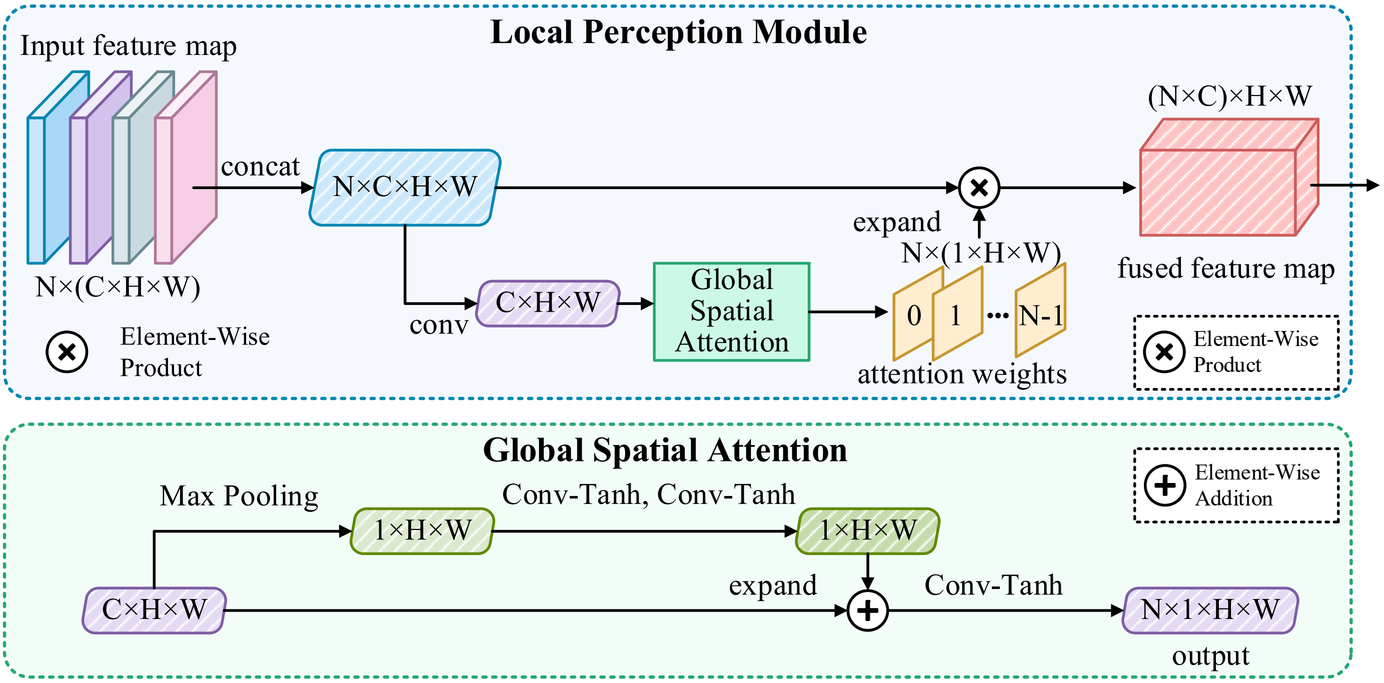


Figure 3. The overall structure of Local Perception Module.

The core of LPM is to solve the problem of local detail loss caused by the sparsity of LiDAR point clouds. In fields such as autonomous driving and robotics, LiDAR is a key sensor for acquiring three-dimensional information about the environment. However, due to factors such as object proximity, surface material, and scanning angle, the density distribution of LiDAR point clouds is very uneven, and the point clouds can be very sparse, especially in distant or occluded areas. Specifically, let $P = \{p_i\}_{i=1}^{N}$ represent a point cloud consisting of N points, where the features of the points are $F = \{F_{p_i}\}_{i=1}^{N}$.

The LPM structure mentioned above is shown in Algorithm 1. First, the feature F is input, and then concatenated to obtain F'. Then, convolve F' to obtain F_1 and enter the global spatial attention. Then, F_1 is maxpooled to obtain F_2. After two *ConvTanh* processes, F_2'' is expanded and added to F_2. Next, F_3 goes through ConvTanh to obtain the final output. Finally, output is expanded and multiplied to obtain the fused feature map, achieving local perception and fusion of features.

Algorithm 1 Local Perception Module

```
Input $F$
$F' = \mathrm{Concat}(F)$
$F_1 = \mathrm{Conv}(F')$
Global Spatial Attention:
    $F_2 = \mathrm{MaxPooling}(F_1)$
    $F_2' = \mathrm{ConvTanh}(F_2)$
    $F_2'' = \mathrm{ConvTanh}(F_2')$
    $F_3 = \mathrm{Expand}(F_2'') + F_2$
    output = ConvTanh(ConvTanh($F_3$))
    Return output
expanded_weights = Expand(output)
fused =WiseProduct($F'$, expanded_weights)
Output fused
```

3.3. Inter-Channel Correlation Module Based on Channel Similarity

For traditional LiDAR point cloud clustering module works by first dividing the point clouds in the offset space into subsets based on the semantic labels of the LiDAR point clouds, which contain only point clouds with the same semantics. The subsets are further grouped using a clustering algorithm. Ultimately, each LiDAR point cloud subset obtained after clustering is considered as an instance. That is, the points in each LiDAR point cloud subset have the same instance label.

However, some post-processing operations aimed at improving the segmentation accuracy result in getting more duplicates or merging invalid 3D information when clustering. This results in increased network computation and poor segmentation network performance in complex scenarios. Therefore, a new clustering module, Inter-Channel Correlation Module (ICCM), is proposed in this paper. The structure of ICCM is shown in Figure 4. LiDAR point cloud data is usually high-dimensional, sparse, and noisy, which makes instance segmentation very challenging. The ICCM module weights the temporal, spatial, and channel dimensions to help the LCPSNet model focus on key regions and features, thus improving the accuracy and robustness of segmentation.

As shown in Figure 4, the spatial feature attention module is first utilized to process the LiDAR point cloud data because point cloud data naturally has a spatial structure. Each point in the laser point cloud has corresponding spatial coordinates, and by weighting these spatial features, it can help the model focus on the most important spatial regions. The sphere query algorithm is first used to identify points within a specified spherical neighborhood. Then, the acquired point set context information is encoded and processed and converted into preliminary instance features. The temporal attention module is then utilized to process the temporal data. The successive sweeps of the LiDAR consist of multiple time steps, and the temporal attention module helps the model to learn which time points are important at different time steps. Instance features that incorporate temporal information are output by calculating Q, K, and V to weight the temporal dimension of the processed point cloud data. Finally, the channel attention weights are assigned by calculating the channel similarity of the instance features to extract the effective channel features and release the redundant ones.

Specifically, for each point selected by the farthest point sampling, a sphere with a unit ball of R is defined with its center. Subsequently, the individual points of the original point cloud are filtered to cull out all points within that sphere that are not more than R away from the center point, which are considered as neighborhood points. If the number of point clouds within a cluster of neighboring points is more than a set threshold B, the closest B neighboring points are selected as local features in order of distance. If the number of neighboring points is less than B, repeat the filling process for these points to ensure that the number reaches B. These B neighboring points are collected as local point cloud features $F_{local} \in R^{B\times D}$, where D denotes the number of channels. The relative coordinates between the sampled points and the neighboring points are also calculated and normalized by the neighborhood radius r to form the local coordinates P_{local}. The initial instance features (Feature map 3) are finalized by weighted averaging of local features and local coordinates. As shown in Equation 1.

$$Feature\ map\ 3 = Feature\ map\ 1 \times \frac{F_{local} + P_{local}}{2} \tag{1}$$

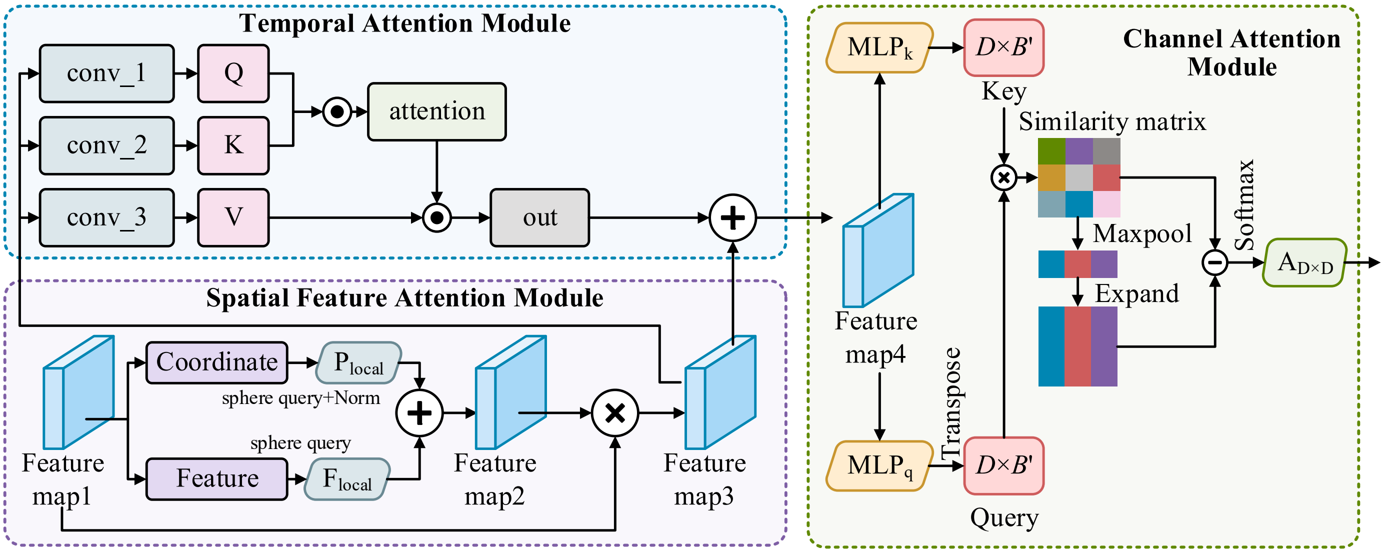


Figure 4. The structure of the ICCM.

The temporal attention module computes the temporal attention of the input feature graph. It computes the Query, Key and Value tensor and computes the attention weights using the dot product attention mechanism. The Query, Key, and Value tensor are shown in Equations 2, 3, and 4.

$$Q = W_q \cdot Feature\ map\ 3 \tag{2}$$

$$K = W_k \cdot Feature\ map\ 3 \tag{3}$$

$$V = W_v \cdot Feature\ map\ 3 \tag{4}$$

Then, the attention fraction is obtained as shown in Equation 5.

$$Attention = softmax(QK^T) \tag{5}$$

The values are weighted and summed using the attention score, as shown in Equation 6. The final feature map 4 to be input into the channel attention is the instance feature F_{ins} in the LiDAR point cloud instance segmentation task.

$$F_{ins} = Feature\ map\ 4 = \gamma \cdot (V \cdot Attention^T) + Feature\ map\ 3 \tag{6}$$

where W_q, W_k, and W_v are the weights of the convolutional layers used for query, key, and value computation, γ is the learnable parameter, and softmax computes softmax along the time dimension.

A transpose operation is performed on F_{ins} to change the shape. Two MLPs are then used to manipulate each channel's feature vector $C_i \in \mathbb{R}^N$ with $C_i = R^N, F_{ins} = [C_1, C_2, \ldots, C_D]$ to encode the contextual information of each channel feature vector. Thereby, the original B points are randomly replaced with fewer B' points, where $B' = B/r_{ratio}; r_{ratio} > 1$ is the scale parameter, and the value is taken as 4 here. Compared with obtaining all points or points extracted under other constraints,

doi:10.20944/preprints202508.0572.v1

this operation can effectively reduce the number of features at the origin and fully preserve the original information. As shown in Equations 7 and 8.

$$q_i = \mathrm{MLP}_q(C_i) \tag{7}$$

$$k_i = \mathrm{MLP}_k(C_i) \tag{8}$$

where MLP_q and MLP_k are the two MLPs operating on the query matrix and the key matrix. Synthesizing the above yields Equation 9 and Equation 10.

$$Q_{B' \times D} = [q_1, q_2, \dots, q_D] \tag{9}$$

$$K_{B' \times D} = [k_1, k_2, \dots, k_D] \tag{10}$$

The corresponding channel similarity matrix is subsequently computed by dot-producting the transpose of the query matrix with the key matrix, which can be expressed as Equation 11.

$$S_{D \times D} = Q^T K \tag{11}$$

where Q is the query matrix and K is the key matrix.

S_{ij} is the similarity between the ith channel and the jth channel of the F_{ins} feature map and an element of the channel similarity matrix. The obtained similarity matrix (S) between the channels is maximally pooled to obtain the column direction expansion, which results in a weight matrix of the same size as the original matrix. By subtracting the weight matrix from the similarity matrix, similar features are retained while redundant features are removed. The refined version of the weight matrix D×D is derived by multiplying any value with the value matrix via Softmax's activation function. This design ensures that the weights optimized for the other channels are de-redundant between each channel. The accumulation of invalid information in the original features is effectively circumvented, thus enhancing the uniqueness and accuracy of the overall feature expression. The process of calculating the weight matrix can be expressed as Equation 12.

$$A_{D \times D} = \mathrm{Softmax}\left\{\underset{1\to D}{\mathrm{expand}}\left[\underset{1\to D}{\mathrm{Maxpool}}(S)\right] - S\right\} \tag{12}$$

Based on the weight matrix, the instance feature channel information is refined by performing a weighted sum over all channels, and another MLP operation is applied to obtain the weight matrix V. Specifically, this can be expressed as Equation 13 and Equation 14.

$$V_{B \times D} = [v_1, v_2, \dots, v_D] \tag{13}$$

$$v_i = \mathrm{MLP}(C_i); v_i \in \mathrm{R}^B \tag{14}$$

Multiplying the value matrix V with the weight matrix achieves the same effect as refining the instance feature channel. In addition, the training is simplified by using residual concatenation with learnable weights α. The processed instance features are shown in Equation 15.

$$F'_{ins} = F_{ins} + \alpha \cdot VA \tag{15}$$

3.4. Cross Entropy Loss Function

In LiDAR point cloud instance segmentation, in order to supervise the point-by-point semantic prediction, a weighted multi-class cross entropy loss $\mathcal{L}_{\mathrm{CE}}$ is used for the set of valid points, as shown in Equation 16. The predicted probability of each point is compared with its true semantic label and averaged as semantic loss. Invalid/filler points are ignored during training, and weights can be set by category frequency to mitigate category imbalance. If the labeling is noisy, light label smoothing can be added.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\sum_{c=1}^{C} w_c \; y_{i,c} \; \log p_{i,c} \tag{16}$$

where $y_{i,c}$ is the one-hot truth value, w_c is the optional category weight to mitigate the class imbalance, and $p_{i,c}$ is the probability of each point to the C semantic categories obtained by picking up MLP and softmax at the output F'_{ins}.

4. Experiments

4.1. Datasets and Metrics

The SemanticKITTI dataset [29] is a large-scale dataset constructed on the original KITTI dataset for 3D LiDAR point cloud segmentation. Its core contribution is to provide dense, point-by-point semantic and instance annotations for dynamic outdoor scenes for KITTI. Based on data from the KITTI Odometry Benchmark, it provides point-by-point annotations for all 22 of these sequences (sequences 00 to 21), totaling over 43,000 LiDAR scans in length. This means that it is a dynamic dataset containing temporal information. Unlike the sparse bounding box of KITTI, SemanticKITTI provides labels for all points in each scan, truly enabling a comprehensive understanding of the entire 3D space. It contains a total of 28 categories, 19 of which are used for evaluation, covering almost all elements in an autonomous driving scenario. The examples of SemanticKITTI dataset are shown in Figure 5.

The Waymo dataset [30] is a large-scale LiDAR point-cloud benchmark for autonomous driving. Data were collected by a fleet equipped with five LiDAR sensors and five high-resolution RGB cameras, operating day and night in cities such as San Francisco. The dataset comprises 1,150 scenes, each 20 seconds long, yielding 230,000 frames in total. Range images and RGB images are synchronized and released at 10 Hz. Among these scenes, 1,000 are designated for training and 159 for testing. The examples of Waymo dataset are shown in Figure 6.

Figure 5. The examples of SemanticKITTI dataset.

Figure 6. The examples of Waymo dataset.

Instance segmentation is essentially a subset of panoramic segmentation. Using metrics for panoramic segmentation provides a more comprehensive view by evaluating not only the performance of instance segmentation but also the semantic segmentation capability that underlies it. Accordingly, we evaluate semantic segmentation using mean Intersection over Union (mIoU). This standard metric quantifies the overlap between predictions and ground truth. For each class we compute IoU as intersection over union and then average the classwise IoUs. As a point level measure, mIoU captures semantic classification accuracy and is suitable when the evaluation focuses only on semantic information.

PQ metrics were initially used for image panorama segmentation evaluation and have been gradually introduced to the 3D point cloud domain in recent years. PQ integrates the recognition quality (RQ) and segmentation quality (SQ) of object instances and is able to evaluate the instance recognition ability and segmentation accuracy of the model simultaneously. Its definition is shown in Equation 17.

$$PQ = \frac{\sum_{TP} IoU}{|TP| + \frac{1}{2}|FN| + \frac{1}{2}|FP|} \tag{17}$$

where TP denotes the number of true positive cases, and FN and FP are the number of false negative and false positive cases, respectively. The numerator part denotes the sum of IoUs of all TPs, and the denominator is the weighted sum of detection and segmentation errors.

4.2. Ablation Experiments

In this paper, ablation experiments are conducted on SemanticKITTI dataset and Waymo dataset, respectively, to verify the effectiveness of each module in LCPSNet. Since the LiDAR point cloud possesses natural defects such as sparsity, inhomogeneous density, and viewpoint dependence, it puts higher requirements on the model's local perception ability and global semantic modeling ability. Therefore, the design of the modules should not only improve the point-level semantic prediction accuracy but also enhance the instance differentiation and structural consistency modeling ability.

First, mIoU and PQ on SemanticKITTI dataset are shown in Table 1, and IoU on each category is shown in Table 2. where mIoU is a weighted average of the weights by category, which is not equal to the simple average.

From Table 1, it can be seen that when no module is introduced, the overall performance of Baseline is relatively weak, the mIoU is only 69.9%, and the PQ value is 61.4%, which indicates that there is much room for improvement of the model in terms of semantic recognition and instance matching accuracy. After the introduction of the LPM module, the mIoU increases to 72.4%, and the PQ increases to 64.8%. This shows that the module effectively enhances the semantic discrimination of local spatial regions in the LiDAR point cloud, and LPM especially improves the segmentation accuracy of dynamic object categories. In particular, LPM improves the segmentation accuracy of dynamic object categories, such as ped from 73.7% to 75.8%, walk from 75.6% to 76.4%, and truc from 59.1% to 60.8%. This validates its ability to recognize fine-grained semantic boundaries.

Table 1. Ablation experiments of Different Module Combinations on SemanticKITTI Dataset.

Combination	LPM	ICCM	mIoU (%)	PQ
Baseline	×	×	69.9	61.4
LPM	√	×	72.4	64.8
ICCM	×	√	72.9	65.2
LCPSNet (Ours)	**√**	**√**	**77.1**	**70.9**

Furthermore, when the ICCM module is introduced alone, the mIoU is 72.9% and the PQ reaches 65.2%, which achieves a significant improvement on several structural or low sample categories. ICCM enhances the model's ability to model structurally complex regions by capturing long-range dependencies and reinforcing semantic consistency among point clouds. For example, m.list increases from 56.1% to 58.1%, terr from 68.1% to 69.6%, and sign from 64.9% to 65.9%. This shows that ICCM has a significant advantage in modeling global contextual semantic consistency, which is especially suitable for improving the recognition accuracy of small targets and fuzzy boundary regions.

When the two modules are jointly introduced to construct the full LCPSNet model, the performance is optimized, with the mIoU reaching 77.1% and the PQ improving to 70.9%, which is 7.2% and 9.5% compared to the baseline, respectively. In Table 2, it can be seen that the model achieves the highest segmentation accuracy in most of the classes. For example, it is significantly ahead in the ground static categories such as build (93.8%), veg (87.8%), and trun (76.6%), as well as the complex m.list (60.9%) and walk (79.7%) categories. This indicates that the synergy of the two modules effectively mitigates the common problems of uneven density, semantic boundary blurring, and category imbalance in LiDAR point clouds.

Table 2. Ablation experiments of Different Module Combinations on SemanticKITTI Dataset.

Category	**Baseline**	**LPM**	**ICCM**	**LCPSNet (Ours)**
car	94.3	94.7	95	98.2
bicy	68.3	69.3	68.5	72.4
moto	70.8	72.3	72.8	75.7
truc	59.1	60.8	60.2	63.9
o.veh	69.4	71	71.8	74.5
ped	73.7	75.8	76.1	79.3
b.list	70.5	71.3	71.5	75.2
m.list	56.1	58.1	58.1	60.9
road	88.2	89.2	89.2	92.9
park	69.9	72	71.9	74.4
walk	75.6	76.4	76.8	79.7
o.gro	42.5	44	43.8	46.5
build	89.9	90.1	91	93.8
fenc	67.4	69.7	69.5	72.9
veg	83	84.8	85.3	87.8
trun	72.4	73	73.6	76.6

terr	68.1	70.3	69.6	73.4
pole	63.9	65.1	66.1	68.7
sign	64.9	65.6	65.9	68.9
mIoU	**69.9**	**72.4**	**72.9**	**77.1**

It is worth emphasizing that the significant enhancement of the PQ metric, as an important indicator for panoramic segmentation evaluation, reflects the combined enhancement of the model in terms of instance separation and prediction accuracy. Unlike mIoU, which focuses more on point-level classification, PQ also considers the match between real and predicted instances, including IoU and target overlap determination. Its significant enhancement thus indicates that LCPSNet not only improves the semantic recognition capability but also strengthens the consistent modeling of target boundary and instance structure when dealing with point cloud panorama segmentation tasks.

Both LPM and ICCM provide complementary capabilities in different dimensions of point clouds. The former focuses on local spatial detail enhancement, while the latter strengthens cross-regional semantic consistency. Together, they can significantly improve the segmentation robustness and generalization ability of the model in LiDAR point cloud scenarios with sparse, heterogeneous, and multi-scale structures.

To further validate the robustness and generalization ability of the proposed module under different LiDAR point cloud scenarios, we conduct the same ablation experiments on the Waymo Open Dataset with a larger field of view, higher density, and more complex traffic environments, and the results are shown in Table 3. Compared with SemanticKITTI, the Waymo dataset contains richer urban traffic scenarios, sampled point clouds under different time and weather conditions, and is more challenging. Therefore, this experiment better reflects the adaptability of the module under different point cloud distribution conditions.

From the results in the table, it can be seen that the mIoU of the base model on this dataset is 62.7%, which is slightly lower than that on SemanticKITTI, reflecting the challenge of target class distribution and environment complexity on the performance of the base model in the Waymo dataset. When the LPM module is introduced, the mIoU increases to 66.9%, which is 4.2% higher than that of the baseline. This indicates that LPM can effectively enhance the local feature expression and improve the ability to discriminate the detailed targets when facing the point cloud data with both dense scenes and sparse structures.

Table 3. Ablation Study of Different Module Combinations on Waymo Dataset.

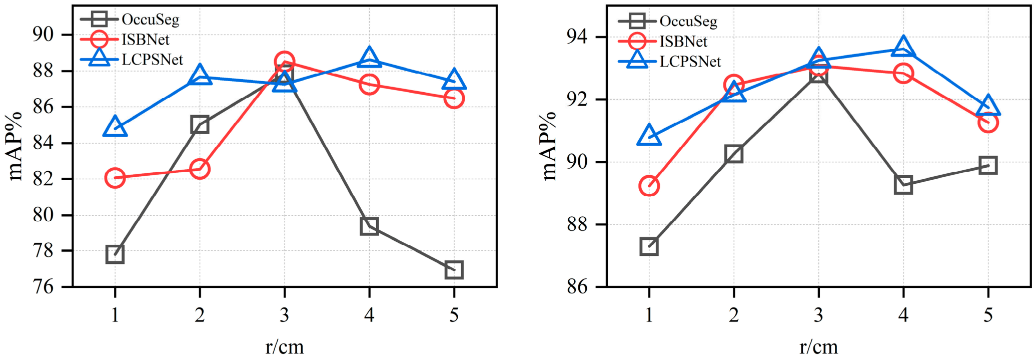
Combination	LPM	ICCM	mIoU (%)
Baseline	×	×	62.7
LPM	√	×	66.9
ICCM	×	√	67.5
LCPSNet (Ours)	**√**	**√**	**70.4**

Similarly, when the ICCM module is introduced alone, mIoU improves to 67.5%, an increase of 4.8%. This reaffirms the effectiveness of ICCM in semantic consistency modeling, especially in the Waymo dataset. Since there are often fuzzy boundaries and semantic overlaps between categories such as lanes, buildings, etc., ICCM can effectively align the contextual semantics to improve prediction consistency. In deep learning, different feature channels often represent different information. The channel attention mechanism allows the model to dynamically learn and adjust the importance weights of different channels. By calculating the similarity between channels, the model can focus on those feature channels that are most useful for the task at hand, while suppressing those that are redundant or noisy. In the task of instance segmentation of LiDAR point clouds, this means that the model can focus more intelligently on the key features that distinguish different objects.

In our full model LCPSNet , LPM is used in conjunction with ICCM, and the mIoU is finally increased to 70.4%, which is 7.7% higher than the base model and further increased by 2.9%-3.5% on top of the two modules. This shows that the two modules still have strong complementarity in the more challenging point cloud scenarios of Waymo, and the joint modeling can capture the local and global features of the point cloud more effectively, which significantly improves the overall performance of the model.

Overall, the experimental results on the Waymo dataset not only further validate the effectiveness of LPM and ICCM in different scenarios. It also shows that the method in this paper has good cross-dataset migration capability and generalization performance and is suitable for real LiDAR semantic sensing tasks in larger-scale and more complex environments.

Figure 7 shows the performance trends of the three LiDAR point cloud segmentation methods on the Waymo and SemanticKITTI datasets for different ball query radius settings. The ball query radius is an important parameter for constructing local neighborhoods in point cloud feature extraction, and its size directly affects the geometric perceptual range and semantic representation effect of the aggregated region. This experiment systematically analyzes the sensitivity and adaptability of OccuSeg, ISBNNet, and LCPSNet methods to this hyperparameter on the Waymo dataset by adjusting the value of the radius r.



(a) On the Waymo Dataset

(b) On the SemanticKITTI Dataset

Figure 7. Influence of ball query radius on mAP.

As can be seen in Figure 7 and Table 4, traditional clustering-based approaches, such as OccuSeg and ISBNNet, are highly dependent on aggregation size. When the radius is small, these methods can only capture local details, and it is difficult to obtain the complete structure and context information of the target, especially when dealing with large targets or complex shapes. When the radius is too large, it is easy to introduce cross-instance interference, resulting in feature mixing and boundary blurring, which affects the accuracy of instance recognition. Therefore, the performance of such methods fluctuates greatly and lacks stability.

Table 4. Effect of different ball radius on mAP for different methods on Waymo dataset and SemanticKITTI dataset.

	r (cm)	1	2	3	4	5
Waymo	OccuSeg	77.79	85.02	87.81	79.37	76.93
	ISBNet	82.07	82.55	88.5	87.24	86.47
	LCPSNet	**84.78**	**87.64**	**87.25**	**88.61**	**87.39**
	r (cm)	1	2	3	4	5
SemanticKITTI	OccuSeg	87.29	90.26	92.82	89.26	89.89
	ISBNet	89.24	92.47	93.08	92.84	91.26
	LCPSNet	**90.78**	**92.16**	**93.25**	**93.61**	**91.74**

In contrast, LCPSNet has stronger scale adaptation capability. By introducing local feature enhancement and a semantic consistency modeling mechanism, the method is able to maintain robust expression ability under different ball query ranges, avoiding performance jitter due to radius changes. In the whole test range, the mAP of LCPSNet on the Waymo dataset is consistently above 85%, and the mAP of LCPSNet on the SemanticKITTI dataset is consistently above 91%. Meanwhile, the fluctuation of LCPSNet's mAP with sphere radius is minimized on both datasets, which fully proves its good structural generality and practical deployment stability.

Ball query radius, as a key structural parameter in the point cloud aggregation process, has a significant impact on segmentation performance. Traditional clustering-based methods are prone to unstable performance under the change of this parameter, indicating their dependence on the modeling of feature space structure. LCPSNet, on the other hand, effectively mitigates this problem through the multi-scale fusion and semantic guidance mechanism, reflecting stronger adaptability and better LiDAR point cloud understanding.

Figure 8 and Figure 9 show the visualization results on SemanticKITTI and Waymo dataset, respectively. In perception systems for autonomous driving, LiDAR is known for its ability to provide accurate 3D spatial information. However, the point clouds it generates are inherently sparse and lack color and texture details. These two characteristics aptly define the need for complementary LPM and ICCM modules.

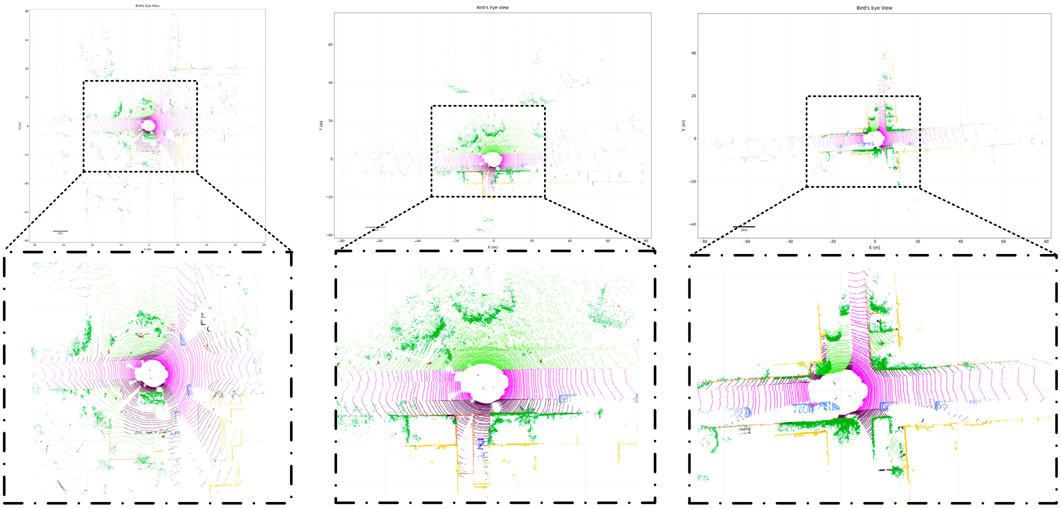


Figure 8. Visualization results for LCPSNet on SemanticKITTI dataset.

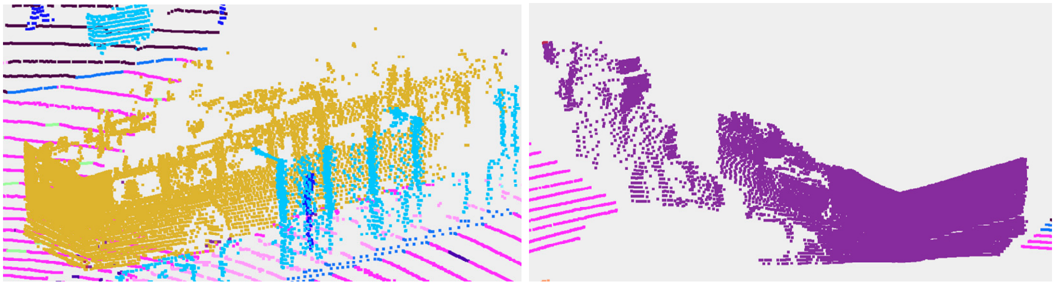


Figure 9. Visualization results for LCPSNet on Waymo dataset.

LPM provides the geometric skeleton for the network. It ensures that the underlying spatial structure of even poorly characterized objects due to distance or few scanning points can be better captured and understood by the network by enhancing the learning of local geometric features. For example, when a car whose color blends in with the environment (with blurred image features) but whose LiDAR point cloud clearly outlines the 3D structure, LPM becomes the key to identifying the object. It enhances the robustness of LCPSNet to the geometric properties of the LiDAR data itself.

ICCM, on the other hand, provides perceptual focus. It utilizes the rich texture and color information of the image to direct the network's attention to those details that are easily overlooked in the LiDAR point cloud. In particular, it directs the network's attention to key details that are easily overlooked in sparse point clouds, such as the contours of pedestrians or the patterns of traffic signs.

Therefore, LCPSNet is not just a simple multimodal fusion network but a deeply complementary system. It combines the robustness originated from LPM for LiDAR geometric information and the acuity originated from ICCM for image perception information. This enables the most accurate judgment to be made for optimal detection performance in difficult scenarios where either LiDAR data is dominant or image data is dominant.

4.3. Comparison Experiments

To fully evaluate the performance of LCPSNet as described in this paper in the LiDAR point cloud semantic segmentation challenge, we compare it with a number of comparable techniques. The results are displayed in Table 5. Since top-down methods are not mainstream in outdoor panorama segmentation, they are difficult to handle backgrounds without fixed shapes and are not competitive. Therefore, in this paper, we only compare with bottom-up methods.

Early methods such as PointNet and PointNet++ only achieve PQs of 17.5 and 20.8 and mIoUs of 18.2 and 23.4, respectively. This indicates that their structures are difficult to model effective spatial context relations when facing complex point cloud geometries, resulting in severely limited segmentation performance. Some subsequent improved methods, such as SSCNS, PolarNet, and PointGroup, introduce sparse convolution, polar coordinate transformation, or clustering mechanisms in the feature extraction strategy, which leads to improved performance. For example, PolarNet achieves a PQ of 54.3 and an mIoU of 55.7, demonstrating progress in spatial structure alignment. However, PointGroup achieves a PQ of 41.7 with an mIoU of 42.5, reflecting its shortcomings in instance boundary modeling.

Table 5. Comparison results between LCPSNet and other methods on SemanticKITTI dataset.

Method	PQ	mIoU (%)
PointNet	17.5	18.2
PointNet++	20.8	23.4
SSCNS	35.2	37.9
PolarNet	54.3	55.7
PointGroup	41.7	42.5
Cylinder3D	66.8	68.9
AF2S3Net	64.9	69.7
RangeFormer	64.1	73.6
SDSeg3D	62.6	70.4
SpAtten	70.5	76.8
LCPSNet (Ours)	**70.9**	**77.1**

Mainstream high-performance methods such as Cylinder3D, AF2S3Net, RangeFormer, and SDSeg3D achieve high scores in both metrics. Among them, Cylinder3D reaches 66.8 in PQ and 68.9 in mIoU, which reflects the modeling advantages of columnar structures in dealing with dense voxels and ground targets. AF2S3Net has a PQ of 64.9 and an mIoU of 69.7, relying on its ability to model semantic spatial relations to achieve a balanced performance. RangeFormer, with the introduction of the Transformer architecture, achieves a higher result of 73.6 in mIoU, but with a PQ of 64.1, suggesting that its global semantic enhancement is effective but still needs to be optimized for instance, matching at the target level. The current best performer is SpAtten with a PQ of 70.5 and mIoU of 76.8.

The LCPSNet proposed in this paper achieves the current best level in both PQ and mIoU, which are 70.9 and 77.1, respectively. Compared with the existing methods, LCPSNet performs better in both semantic consistency and instance separation ability. In conclusion, LCPSNet achieves the simultaneous improvement of semantic accuracy and panoramic quality in the LiDAR point cloud semantic segmentation task, which verifies its excellent performance in the real scene perception task.

5. Conclusions

Aiming at the problems of information loss due to voxelization processing and invalid feature redundancy due to post-processing operations after feature enhancement in LiDAR point cloud instance segmentation network, this paper designs and implements a LiDAR point cloud instance segmentation method (LCPSNet) based on local perception and channel similarity. Based on the traditional FPN that only does prediction or simple cascading in each layer, we propose a location-consistent multi-scale fusion for LiDAR. First, the fusion features of 1/2, 1/4, 1/8, 1/16, and 1/32 are obtained by up-sampling and summing the FPN layers from top to bottom. Subsequently, the 1/16, 1/8, and 1/4 layers are uniformly resampled to the same target resolution and input into the LPM together with the 1/2 layer. Position-by-position spatial weight modeling and cross-scale modulation on the same BEV/polar coordinate unit. This unified raster and position-by-position fusion breaks through the limitations of conventional FPNs that only fuse within scales, enabling deep, large sensory field contexts to synergize with shallow, fine boundaries at the same location. Distant/large targets gain contextual enhancement, while close/small targets retain fine contours. This significantly improves the separability of neighboring instances in sparse, occluded, and scale-varying scenes with little additional computation.

Local Perception Module (LPM) and Global Spatial Attention (GSA) are used after the backbone of LiDAR point cloud instance segmentation to specifically address cross-scale aliasing and neighboring instance sticking due to LiDAR's distance sparsity and scale variation. The LiDAR point cloud is first projected to Range-view and FPN to get N sets of local multi-scale features. GSA guides one of the features to do spatial pooling and convolution to get the global spatial saliency map. The other branch is unfolded and convolved in the group dimension to produce group correlation responses. The two branches are summed to form weights for each group and each position. LPM then broadcasts this weight over the channel, multiplies it element-by-element with the corresponding group of features, and then splices it into a fused feature map in the channel dimension. This is then fed to the semantic and offset header with the instance decoder. LPM and GSA do position-dependent cross-scale modulation on the same polar grid, fusing the deep large sensory field context with the shallow fine boundary alignment, significantly improving the separability of neighboring instances.

In addition, the network innovatively proposes an Inter-Channel Correlation Module (ICCM) based on channel similarity. ICCM models and explicitly de-redundantly models the timing and channels of the point cloud simultaneously. Based on the local geometry obtained from the sphere query, the temporal attention is first used to adaptively select the critical temporal information from multi-frame LiDAR to form robust instance features. Subsequently, the channels are weighted based on channel similarity to suppress redundancy and highlight discrimination, together with learnable channel compression and residual refinement. Compared with the methods relying on post-processing clustering, it is more accurate, robust, and efficient in sparse, noisy, and occluded scenes. Experimental results show that LCPSNet outperforms many existing classical and advanced methods on two mainstream datasets, SemanticKITTI and Waymo, especially when dealing with dense and complex outdoor traffic scenes. Its PQ of 70.9 and mIoU of 77.1 on the SemanticKITTI dataset achieve SOTA performance. In addition, the proposed network is more stable to parameter variations and more generalizable while enhancing local perception and improving segmentation performance.

References

1. Huang X S, Mei G F, Zhang J, et al. A comprehensive survey on point cloud registration [EB/OL]. (2021-03-03) [2024-12-12]. https://arxiv.org/abs/2103.02690v2.
2. Zeng Y H, Jiang C H, Mao J G, et al. CLIP2: contrastive language-image-point pretraining from real world point cloud data[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 17-24, 2023, Vancouver, BC, Canada. New York: IEEE Press, 2023: 15244-15253.
3. Marinos, V., Farmakis, I., Chatzitheodosiou, T., Papouli, D., Theodoropoulos, T., Athanasoulis, D., & Kalavria, E. (2025). Engineering Geological Mapping for the Preservation of Ancient Underground Quarries via a VR Application. *Remote Sensing*, *17*(3), 544.
4. Qian R, Lai X, Li X R. 3D object detection for autonomous driving: a survey[J]. Pattern Recognition, 2022, 130: 108796.
5. Lee S, Lim H, Myung H. Patchwork: fast and robust ground segmentation solving partial under-segmentation using 3D point cloud[C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 23-27, 2022, Kyoto, Japan. New York: IEEE Press, 2022: 13276-13283.
6. Xiao A R, Yang X F, Lu S J, et al. FPS-Net: a convolutional fusion network for large-scale LiDAR point cloud segmentation[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2021, 176: 237-249.
7. Hafiz A M, Bhat G M. A survey on instance segmentation: state of the art[J]. International Journal of Multimedia Information Retrieval, 2020, 9(3): 171-189.
8. Guo Y L, Wang H Y, Hu Q Y, et al. Deep learning for 3D point clouds: a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(12): 4338-4364.
9. Lu B, Liu Y W, Zhang Y H, et al. Point cloud segmentation algorithm based on density awareness and self-attention mechanism[J]. Laser & Optoelectronics Progress, 2024, 61(8): 0811004.
10. Ai D, Zhang X Y, Xu C, et al. Advancements in semantic segmentation methods for large-scale point clouds based on deep learning[J]. Laser & Optoelectronics Progress, 2024, 61(12): 1200003.
11. Zhang K, Zhu Y W, Wang X H, et al. Three-dimensional point cloud semantic segmentation network based on spatial graph convolution network[J]. Laser & Optoelectronics Progress, 2023, 60(2): 0228007.
12. Xu X. Research on 3D instance segmentation method for indoor scene[D]. Daqing: Northeast Petroleum University, 2023: 12-13.
13. Cui L Q, Hao S Y, Luan W Y. Lightweight 3D point cloud instance segmentation algorithm based on Mamba[J]. Computer Engineering and Applications, 2025, 61(8): 194-203.
14. Wang W Y, Yu R, Huang Q G. SGPN: similarity group proposal network for 3D point cloud instance segmentation [EB/OL]. (2017-11-23) [2024-12-12]. arXiv:1711.08588.
15. Hou J, Dai A, Nießner M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 4416-4425.
16. Lin K H, Zhao H M, Lv J J, et al. Face detection and segmentation based on improved mask R-CNN[J]. Discrete Dynamics in Nature and Society, 2020, 2020(1): 9242917.
17. Yang B, Wang J, Clark R, et al. Learning object bounding boxes for 3D instance segmentation on point clouds[C]//Proceedings of the 33rd Conference on Neural Information Processing Systems, 2019: 1-10.
18. Han L, Zheng T, Lan X, et al. OccuSeg: occupancy aware 3D instance segmentation[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 2937-2946.
19. Kolodiazhnyi M, Vorontsova A, Konushin A, et al. Top-down beats bottom-up in 3D instance segmentation[C]//2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 3-8, 2024, Waikoloa, HI, USA. New York: IEEE Press, 2024: 3554-3562.
20. Charles R Q, Li Y, Hao S, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space[EB/OL]. (2017-06-07) [2024-12-12]. https://arxiv.org/abs/1706.02413.
21. Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 77-85.
22. Mo K C, Zhu S L, Chang A X, et al. PartNet: a large-scale benchmark for fine-grained and hierarchical part level 3D object understanding[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 909-918.

23. Graham B, Engelcke M, Maaten V D L. 3D semantic segmentation with submanifold sparse convolutional networks[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018: 9224-9232.
24. Jiang L, Zhao H, Shi S, et al. PointGroup: dual-set point grouping for 3D instance segmentation[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020: 4866-4875.
25. Chen S Y, Fang J M, Zhang Q, et al. Hierarchical aggregation for 3D instance segmentation[EB/OL]. (2021-08-06)[2024-12-12]. arXiv:2108.02350.
26. He T, Shen C, Hengel V D A. DyCO3D: robust instance segmentation of 3D point clouds through dynamic convolution[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021: 354-363.
27. Ngo T D, Hua B S, Nguyen K. ISBNet: a 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 17-24, 2023, Vancouver, BC, Canada. New York: IEEE Press, 2023: 13550-13559.
28. Huang X S, Mei G F, Zhang J, et al. A comprehensive survey on point cloud registration [EB/OL]. (2021-03-03)[2024-12-12]. https://arxiv.org/abs/2103.02690v2.
29. Behley J, Garbade M, Milioto A, et al. Semantickitti: A dataset for semantic scene understanding of lidar sequences[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 9297-9307.
30. Sun P, Kretzschmar H, Dotiwalla X, et al. Scalability in perception for autonomous driving: Waymo open dataset[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2020: 2446-2454.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.