Hindawi
Mobile Information Systems
Volume 2023, Article ID 9872576, 1 page
https://doi.org/10.1155/2023/9872576

Retraction

Retracted: PointLAE: A Point Cloud Semantic Segmentation Neural Network via Multifeature Aggregation for Large-Scale Application

Mobile Information Systems

Received 31 October 2023; Accepted 31 October 2023; Published 1 November 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Li and J. Liu, "PointLAE: A Point Cloud Semantic Segmentation Neural Network via Multifeature Aggregation for Large-Scale Application," *Mobile Information Systems*, vol. 2022, Article ID 9433661, 14 pages, 2022.

Hindawi
Mobile Information Systems
Volume 2022, Article ID 9433661, 14 pages
https://doi.org/10.1155/2022/9433661

Research Article

PointLAE: A Point Cloud Semantic Segmentation Neural Network via Multifeature Aggregation for Large-Scale Application

Jiang Li[1] **and Jinhao Liu**[1,2]

[1]*School of Engineer, Beijing Forestry University, Beijing 100086, China*
[2]*National Engineering Research Centre for Agri-product Quality Traceability, Beijing Technology and Business University, Beijing 100048, China*

Correspondence should be addressed to Jiang Li; 1109076840@qq.com and Jinhao Liu; liujinhao@bjfu.edu.cn

Received 24 March 2022; Revised 21 April 2022; Accepted 29 April 2022; Published 20 June 2022

Academic Editor: Liping Zhang

The fast semantic segmentation algorithm of 3D laser point clouds for large scenes is of great significance for mobile information measurement systems, but the point cloud data is complex and generates problems such as disorder, rotational invariance, sparsity, severe occlusion, and unstructured data. We address the above problems by proposing the random sampling feature aggregation module ATSE module, which solves the problem of effective aggregation of features at different scales, and a new semantic segmentation framework PointLAE, which effectively presegments point clouds and obtains good semantic segmentation results by neural network training based on the features aggregated by the above module. We validate the accuracy of the algorithm by training on Semantic3D, a public dataset of large outdoor scenes, with an accuracy of 90.3, while verifying the robustness of the algorithm on Mvf CNN datasets with different sparsity levels, with an accuracy of 86.2, and on Bjfumap data aggregated by our own mobile environmental information collection platform, with an accuracy of 77.4, demonstrating that the algorithm is good for mobile information complex scale data in mobile information collection with great recognition effect.

1. Introduction

With the rapid development of sensors such as LiDAR, mobile measurement platforms are widely equipped with laser sensors, and it becomes more convenient to obtain point cloud data from different sites. In agriculture and forestry, point clouds are widely used in unattended farmland management, agricultural and forestry operation path planning, urban garden construction planning, and biomass estimation. In the urban environment, it is widely used in 3D modeling of urban buildings and unmanned driving and high-precision urban maps. In aviation, it is widely used in airborne radar high-altitude mapping and flight trajectory planning, etc. Point cloud data is huge and faces traditional data problems such as disorder, rotational invariance, sparsity, severe occlusion, and unstructured data. To solve these problems, researchers have explored point cloud processing methods.

(1) Traditional voxelized deep learning methods and multiview CNN approaches: 3D convolutional neural networks were first applied to identify voxel models by voxelized CNN methods 2015 [1–3]. However, the huge increase in computation and point cloud sparsity after the rasterization of point cloud data limited its development [4]. Li et al. [5] proposed the above model sparsity problem solutions, but difficulties still exist for large amounts of point cloud data. Multiview convolutional neural networks for 3d shape recognition [6] attempt to convert 3D point clouds or shapes into 2D images and classify them using 2D convolutional networks. With the further development of 2D convolutional research, this approach has achieved the best recognition results, but it is difficult to extend this approach to 3D tasks such as mobile acquisition scene resolution and point cloud classification. Meanwhile, the spectral CNNs [7, 8]

method is currently limited to recognizing objects with a similar rich grid structure, and it is unclear how to apply the method to nonshaped objects. Point clouds suffer from problems such as disorder and spatial rotation invariance, and irregular spatial relationships exist when classifying or segmenting. Therefore, existing frameworks for image classification and segmentation cannot be directly applied to point clouds. Many voxel (grid) and multiview convolution-based deep learning frameworks have also been developed with good results. However, these methods inevitably change the original data features, leading to unnecessary data loss, while the computational effort is huge and difficult to adapt to the huge amount of point cloud data.

(2) Point-based deep learning networks: with the development of deep learning, more and more people try to use deep learning algorithms directly on point clouds. Researchers proposed a spatial affine transformation matrix to solve the point cloud rotation by increasing the data dimension and symmetric network structure, and proposed a spatial affine transformation matrix to solve the point cloud rotation problem proposed Pointnet [9] neural network to solve the point cloud disorder problem. Pointnet only performs feature extraction on single. The ability of Pointnet to extract local information of the model is far from enough by only feature extraction of coordinates of a single point. The team proposed the Pointnet++ network [10], which uses fast point sampling (FPS) and multiscale grouping (MSG) to extract local information. However, for this purpose, the scale of these actions is very large. The multiresolution grouping (MRG) method is proposed to fully extract point cloud features to further improve the classification and segmentation. To solve the above problem, a multiscale feature fusion approach is used. A pyramidal multilayer point cloud feature extraction network, Pointsift, was also proposed to solve the point cloud disorder problem and extract indoor point cloud features effectively [11]. A PointCNN network was constructed using a method called "x-transform" to solve the problem that convolutional operations cannot be easily applied to irregular and disordered point cloud data [12]. This kind of network mostly uses the downsampling strategy. Most of the existing algorithms use the downsampling strategy, which is either computationally costly or memory intensive. Currently, this method is widely used. The farthest point sampling requires more than 200 seconds to downsample a point cloud of 1 million points to 10% of the original scale. Point-based deep learning relies on computationally costly kerning or graph construction. Most existing methods have a relatively limited field of perception when extracting features and struggle to efficiently and accurately learn complex geometric structure information from large field point clouds.

(3) Graph-based deep learning networks for point clouds: recent work by researchers has begun to experiment with directly processing large-scale point clouds. For example, SPG [13, 14] uses feature descriptions of large-scale point clouds of attractions, and methods such as FCPN [15] combine the advantages of voxels and points to process large-scale point clouds. Although these methods have achieved good segmentation results, most of them require too much preprocessing computation or memory footprint to be deployed in practical applications. Wang et al. proposed a PCCN [16] network based on a parametric continuous convolution layer. The number of kernels in this layer is parameterized by MLPs and spans a continuous vector space. Hughes et al. proposed a kernel point full convolution network (KP-FCNN) based on kernel point convolution (KPConv) [17]. 3DCNN- DQN-RN [18] is out for efficient semantic analysis of large-scale point clouds. The network uses a 3D CNN network to learn the spatial distribution and color features and then uses DQN to locate class objects. The stitched feature vectors are fed into the residual RNN to obtain the final segmentation results. These methods are still limited by the perceptual field range and are difficult to segment effectively for point clouds with different sparsity scenes.

For the large-scale field point cloud data collected by the telemetry system, it faces the problems of scene complexity, oversized computation, and uneven sparsity in addition to the inherent characteristics of disorder, rotational invariance, sparsity, severe occlusion, and unstructured point clouds. The above methods suffer from insufficient large scale processing, excessive parameter scale, and low segmentation accuracy in practice. Therefore, we propose the point cloud deep learning framework PointLAE, which consists of multiple feature extraction modules ATSE and deep learning network PointLAE. The innovation points are as follows:

(1) For large-scale scene point cloud down sampling problems, the module ATSE is proposed to rapidly downsample in real time and maximize the effective information of retaining the geometric features of the point cloud.

(2) Design a neural network, PointLAE, that effectively predivides the point cloud by ATSE module features, speculates the feeling range of the point cloud based on the over-segmentation results, reduces the memory and time required for operation, and achieves good results for both sparse and dense point clouds.

2. Methods and Materials

In this section, the details of the feature aggregation module (ATSE) are described. First, we introduce the feature computation of the point cloud; then, ATSE feature selection. Finally, PoingLAE, a neural network for the semantic segmentation of a point cloud with high performance, is introduced.

2.1. Feature Ensemble Module ATSE. Aiming at a massive point cloud scene on the order of tens of millions of meters in area, which is collected by mobile laser radar, applying deep learning methods directly on data and entering the deep learning network directly are very difficult tasks. Therefore, effectively reducing the large-scale point cloud is an important procedure. Qi et al. use the strategy of furthest sampling generation in pointnet FPS [10], and Landure and Simonovsky [13] use the voxelization method for preprocessing in Superpoint graph. These methods are computationally large compared with either FPS method. The voxelization method has high memory occupancy of the GPU, and the effect of a regular voxel cutting block on the receptive field of the point cloud has a slow calculation speed. The method of random sampling has a low computational cost, small GPU memory occupancy, and high operational efficiency. Also, the number of inputs to the point cloud is not required, even if any size point cloud can be directly input to the network for training. However, Randle et al. note that the method of random downsampling leads to an absence of valid information for the point cloud. Therefore, we want to construct redundant features of the point cloud to reduce the absence of valid information. First, the entire large-scale point cloud space is randomly downsampled, the K close neighbors of the neighborhood space are found for each point P with the k-nearest neighbor search algorithm KNN, and the geometric features and positions of the close neighbors are encoded with the relative positions. To limit the computational effort, we only define descriptors of four local geometric features that describe the spatial feature structure, linearity, planarity, scattering, and perpendicularity of the point cloud. In each KNN neighborhood, we compute point cloud covariance eigenvalues $\lambda 1 \geq \lambda 2 \geq \lambda 3$, according to the principle of the optimal neighborhood in Weinmann et al., where the neighborhood size selected results in $(\lambda 1/\lambda, \lambda 2/\lambda, \lambda 3/\lambda)$ and is epigenetically minimized [19]. These features allow the geometry of the local neighborhood to be best described through the following vectors.

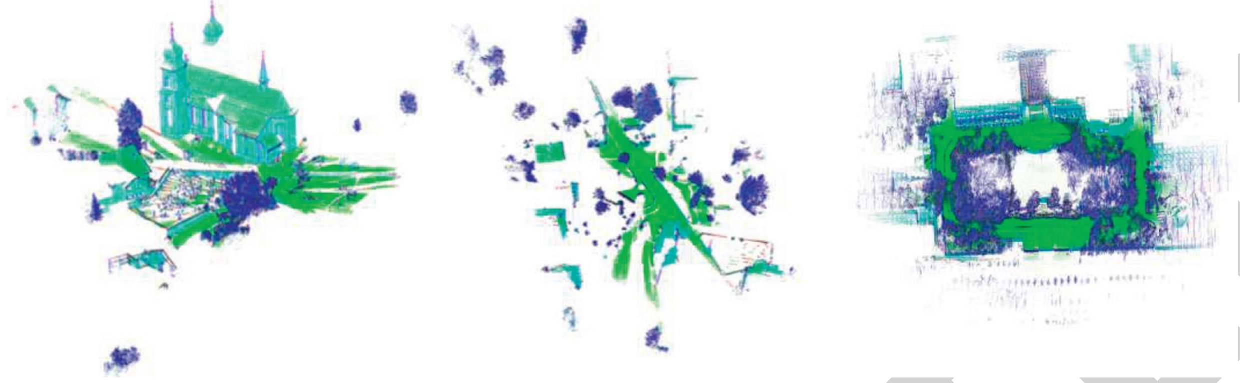
We supplemented the calculation formula for the full variance, which has strong descriptive power for the extent of undulation on the point cloud surface. For example, the full variance is higher for trees and grass than for artificial surfaces and buildings. Perpendicular descriptor: the vertical properties of the best neighborhood are important for distinguishing roads from elevations and between polygons and the cloud of vertical object points that are similar.

$$
\begin{aligned}
Linearity &= \frac{\lambda_1 - \lambda_2}{\lambda_1},\\
Planarity &= \frac{\lambda_2 - \lambda_3}{\lambda_1},\\
Scattering &= \frac{\lambda_3}{\lambda_1},\\
Q &= \sqrt[3]{\lambda_1 \cdot \lambda_2 \cdot \lambda_3}.
\end{aligned}
\tag{1}
$$

The first three feature ensembles are called the dimensionality properties of the point cloud. The vertical feature descriptors are also taken from the above feature vectors and eigenvalues. Let $A1$, $A2$, $A3$ be the difference between the point cloud and the three related eigenvectors $\lambda 1$, $\lambda 2$, $\lambda 3$. The vector of primaries is defined for these three vectors as the sum of coordinate absolute values to which the eigenvectors are weighted by their eigenvalues. The vertical component of this vector is used to characterize the perpendicularity of a KNN point neighborhood. It is convenient to divide the horizontal neighborhood and the linear vertical neighborhood since it reaches its minimum (equal to zero) for the horizontal neighborhood and its maximum (equal to one) for the linear vertical neighborhood. As shown in Figure 1(a), the point cloud geometric feature descriptors are visualized for the semantic3d dataset, where red represents the point cloud linear feature, green represents the planarity feature, and purple represents the full variance trait Q. This described the point cloud surface undulating ability, performing well on low vegetation and high vegetation. As shown, the local feature descriptors can better capture the local geometry of the point cloud. In the figure, cyan represents the descriptor of the vertical characteristics of the point cloud, which better reflects the vertical geometric properties of the point cloud. In addition, laser radar scanning was also performed on campus with the rollout of the Bjfumap dataset shown in Figures 1(b) and 1(c). The linear features in red describe the road boundary and geometric features such as the building boundary. The planar features in green are a good representation of the planar geometric features. The point cloud scattering features in purple represent the full variance feature Q that describes the undulation ability on the point cloud surface, representing the trees, etc. Perform well in representing features such as the building wall. These four features can be used not only in the feature aggregation module shown in Figure 2, but also in the subsequent global optimization hypersegmentation of neural network energy. Linearity: describes the degree of tensile elongation of the neighborhood of the point cloud; Planarity: assesses its fit to the plane; Scattering: corresponds to a spherical neighborhood and describes the characteristics of isotropy.

To construct redundant point cloud features for reducing the missing point cloud information by k-machine downsampling and incorporating the geometric feature descriptors of the point cloud into subsequent deep neural networks, we propose a module for attention aggregation of point cloud features: the ATSE module.

Firstly: Find the nearest K neighborhood points ${}^{i}_{p}$ in the Euclidean space for each point $\{p_i^1 \cdots p_i^k \cdots p_i^K\}$ using the k-nearest neighbor algorithm, and compute the geometric feature descriptors ${}^{i}_{\mathrm{f}}L$, ${}^{i}_{\mathrm{f}}s$, ${}^{i}_{\mathrm{f}}P$, ${}^{i}_{\mathrm{f}}V$, ${}^{i}_{\mathrm{f}}\mathrm{Q}$ of the centre and point of each neighborhood geometric feature descriptor according to equations (1)–(3). Then, the geometric feature descriptor $\mathrm{f}_{i=1}^{i=k}L$, $\mathrm{f}_{i=1}^{i=k}s$, $\mathrm{f}_{i=1}^{i=k}P$, $\mathrm{f}_{i=1}^{i=k}V$, $\mathrm{f}_{i=1}^{i=k}\mathrm{Q}$ of the centre point of each neighborhood and the neighborhood geometric feature of the point are concatenated with MLP sharing weights to obtain a new point feature f_i^k. This feature is a redundant feature of points that can effectively reduce the loss of effective



(a) (b) (c)

Figure 1: (a) Point cloud geometric feature visualization in semantic3d dataset. (b) Main building area point cloud geometric feature visualization in bjfumap dataset. (c) Point cloud geometric feature visualization of Avenue area in bjfumap dataset.

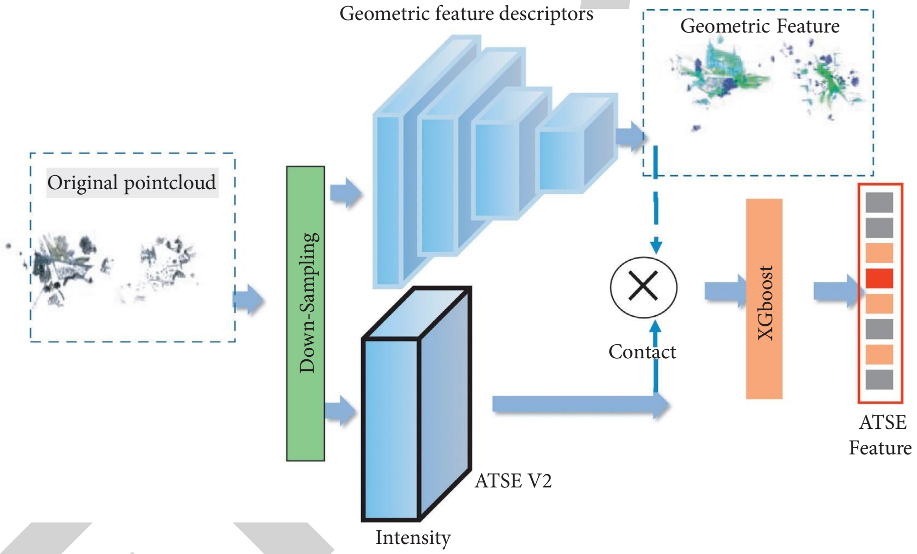


Figure 2: Diagram of the ATSE structure. This module is used for efficient point cloud feature aggregation. ATSE V2 incorporates the reflection intensity information of the point cloud.

information in a random sampling process. By the method of concat, f_i^k is concatenated with the point geometric feature descriptors ${}_{\mathrm{f}}^{i}L$, ${}_{\mathrm{f}}^{i}s$, ${}_{\mathrm{f}}^{i}P$, ${}_{\mathrm{f}}^{i}V$, ${}_{\mathrm{f}}^{i}\mathrm{Q}$ to obtain point features $\widehat{\mathbf{f}}_i^k$.

Secondly: To combine the above features, a max-pooling approach can be used. However, this approach results in a lot of missing useful information, and we hope to automatically learn using the method of attention to select useful information among the features concatenated by aggregation 1. Here, we adopt the approach of xgboost [20] to analyze the contribution of different model features to the results. The contribution values were calculated by shapes. In equation (6), we adopt the function α to learn an attachment score for each feature and a separate attachment score for each point by sharing the learnable parameters of the MLP. Using this software mask that automatically selects the features, a weighted summation at the neighborhood feature point level is obtained, as shown in (2). A module for attentional aggregation of cloud features: ATSE module v1.

$$\begin{aligned} f_i^k &= MLP\left(f_{i=1}^{i=k}L\oplus f_{i=1}^{i=k}s\oplus f_{i=1}^{i=k}P\oplus f_{i=1}^{i=k}V\right),\\ \widehat{f}_i^k &= concat\left(f_i^k, {}_f^iL, {}_f^is, {}_f^iP, {}_f^iV\right),\\ score_i^k &= \alpha\left(\widehat{f}_i^k, W\right). \end{aligned} \tag{2}$$

Thirdly: Point clouds are acquired using many different methods. Currently, laser radar is the main method to acquire point clouds in a large scene environment such as outdoors, in which the laser radar acquires point clouds with an important feature, reflection intensity. We supplemented the above features with the reflection intensity of the point cloud and the laser reflection intensity. Although there is some distance between the point and the scanner, the reflection intensity is mainly influenced by the surface material of the scanned object. This is also an important feature for point cloud classification.

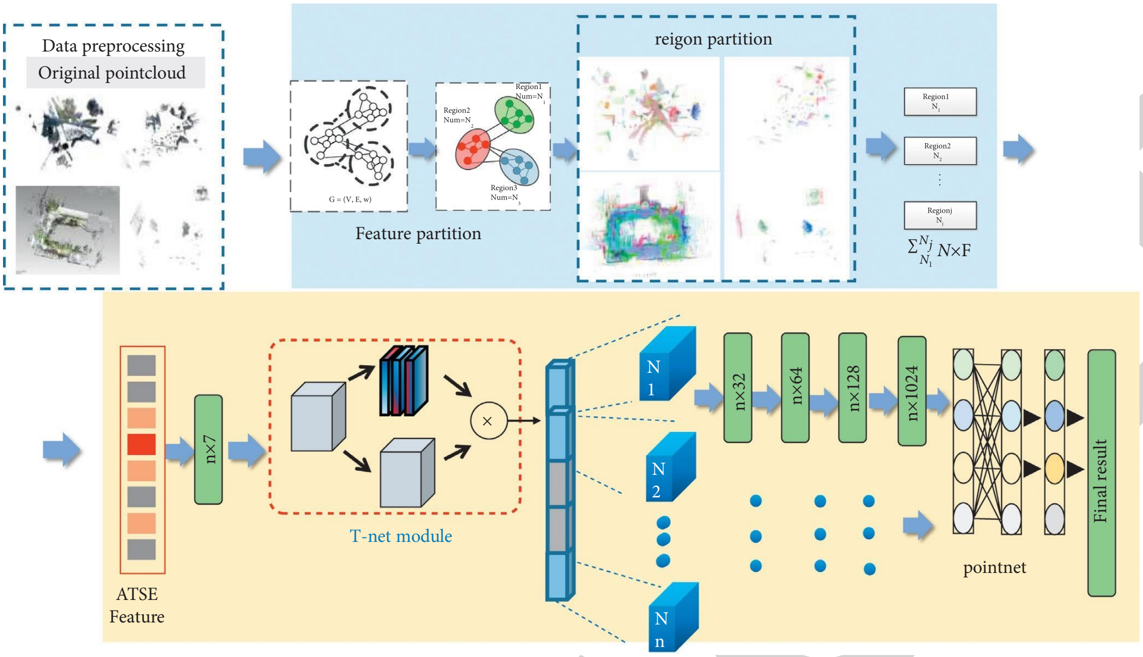


Figure 3: Structure of PointLAE, which has 3 main parts, receives the point cloud multifeature information aggregated by ATSE, and the point cloud oversegmentation based on energy entropy, as well as the proposal of a type of point cloud feeling area according to the oversegmentation results. After going through Pointnet and the fully connected layer, the point cloud is segmented into multiple classes.

$$\begin{aligned} I &= \|Intensity\|,\\ f_i^k &= MLP\left(f_{i=1}^{i=k} L \oplus f_{i=1}^{i=k} s \oplus f_{i=1}^{i=k} P \oplus f_{i=1}^{i=k} V \oplus f_{i=1}^{i=k} Q\right),\\ score_i^k &= \alpha\left(\hat{f}_i^k, W\right),\\ \tilde{f}_i &= \sum_{k=1}^{K}\left(\hat{f}_i^k \cdot score_i^k\right). \end{aligned} \tag{3}$$

Similarly, we find the nearest K neighborhood points in Euclidean space for each point using the k-nearest neighbor algorithm, The point I neighborhood mean calculated the aggregated features characteristic of the fused reflection intensity feature were obtained. We named this module ATSE v2.

We present the point cloud feature aggregation module ATSE, which aggregates features for optimization, where ATSE v2 is computed at ATSE 1, which adds the important information of laser reflection intensity to data acquired by laser radar outdoors, which would be mentioned in the later experiments to effectively improve the accuracy and robustness of point cloud segmentation, but for point cloud data without reflection intensity information, such as image generation, it can be processed by ATSE v1.

We presented the point cloud feature aggregation module ATSE that aggregates features for optimization. ATSE v2, compared to ATSE v1, adds the important information of laser reflection intensity to outdoor data acquired by laser radar ATSE v2 is mentioned in later experiments to effectively improve the accuracy and robustness of point cloud segmentation. However, for point cloud data without reflection intensity information, such as image generation, ATSE v1 can be used for processing.

2.2. Semantic PointLAE Segmentation Framework. As shown in Figure 3, the point cloud is input to PointLAE through a preset block batch, goes through the rotation network T-net for point cloud rotation to eliminate the rotation invariance of the point cloud, and then through the ATSE module to obtain features $\widehat{\mathbf{f}}_i^k$ with D dimension, in addition to the x, y, and z 3D features inherent to the point cloud, is $n\times\ (3+\text{D})$. If processing continues through subsequent modules of the Pointnet at this time, the problem of the receptive field of Pointnet will occur, considering that the point of input is a large-scale scene. Pointnet has difficulty acquiring local geometric features. Therefore, it is essential to significantly increase the receptive field of each point. The overall geometric details of the input point cloud are expected to be preserved.

Uniform segmentation: the first step of the algorithm is to divide the point cloud into simple but meaningful small sections. Our module can generate high-quality point cloud oversegmentation, equivalent to semantically robust presegmentation, with the following properties:

(1) Object nonoverlap: point clouds on different objects are nonoverlapping with each other, especially when the semantics represented are different.

(2) Marginality: hypersegmented clusters of point clouds coincide with boundaries between objects.

The point cloud after downsampling is referred to as an undirected graph $G = (\text{V}, \text{e})$, where V represents a node. We relate its local geometric feature vector $\widehat{\mathbf{f}}_i^k$ and R (dimensionality and perpendicularity, geometric feature descriptors computed in Section 1) and computed the segmentation constants using the graph G constructor. This is defined as a vector $\frac{i}{\widehat{\mathbf{f}}} \in \text{R}^{V\times\text{D}}$ that minimizes the following Potts'

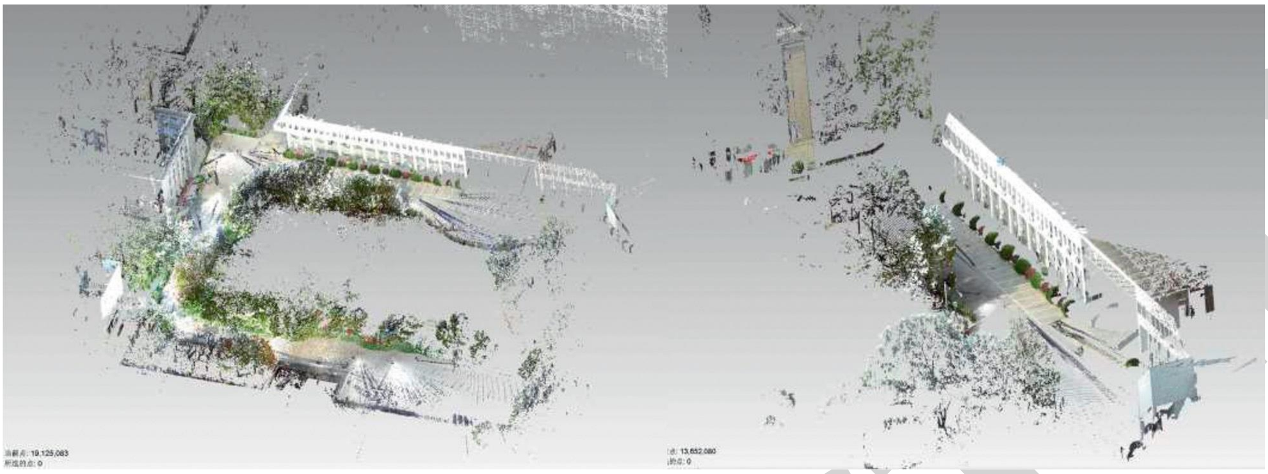


Figure 4: Visualization of the point cloud color map obtained by scanning.

segmentation energies by optimizing the following energy functions [21].

$$g = \underset{g\in R^{D\times V}}{\arg\min} \sum_{i\in V} \left\|g_i - f_i\right\|^2 + \rho \sum_{(i,j)\in E} \delta\left(g_i - g_j \neq 0\right)^{V\times 4}. \quad (4)$$

This energy optimization can be solved with a 0-cut pursuit algorithm proposed by Landrieu [21]. With fast and efficient effects, the granularity of the segments will increase as the point cloud becomes more complex. Figure 4 shows the effect of the optimal hypersegmentation of individual datasets energy. The 64-dimensional point cloud feature shown in Figure 3 is segmented, after (8) energy optimized segmentation, into N point sets of varying sizes, i points within each set. Different from Pointnet++, where the sampling method is multiple stepwise downsampling, here we apply a new method of receptive field size setting. Since the above energy optimized the characteristics of segmentation, each block point cloud is semantically homogeneous. The density of the point cloud within a block is calculated, and the size of the receptive field of the point cloud is determined by calculating the density of the point cloud.

A smaller receptive field is employed in the block where the point cloud is dense to extract complex points. The shape of the cloud geometry is in the block where the point cloud is sparse, a larger receptive field is adopted to better capture the features of the point cloud. Finally, the output feature F is input to Pointnet training and then through the full connection layer to achieve the classification results.

In this subsection, we propose PointLAE, use the results proposed in Section 3 to carry out the point cloud energy entropy based hypersegmentation, and according to the results of hypersegmentation, propose the method of preset point cloud receptive field. This approach can better capture the point cloud local features in different degrees of hydrophobicity in the point cloud, and finally, after the network is trained, can achieve a good segmentation effect.

We segmented the 64-dimensional point cloud feature shown in Figure 3, after energy optimized segmentation, into m point sets of varying sizes, i points within each point set, MLP of shared parameters was done.

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} N_i(x) f_i^{(j)}}{\sum_{i=1}^{k} N_i(x)} \textit{ where } w_i(x) = \frac{1}{N^{(x,x_i)}}, i = n. \quad (5)$$

In each cell block after L hyper-segmentation, they have similar point cloud effective features inside, so deep learning is performed in each. Firstly, the network is designed to go through a T net, and the point cloud is rotated and transformed with the purpose of eliminating point cloud disorder, Then Do MLP with shared parameters, up-dimension to get a 1024-dimensional feature vector, then do max-pooling in each cluster to get 1024-dimensional local features of each cluster, and finally do max-pooling once more for the local features of m clusters $m \times 1024$ to get the global features N. Copy the global features n times, concat the 64-dimensional feature vector on the $n \times 1024$dimensional global features to get $n \times 1088$ point feature values, and after MLP, the point cloud is partitioned into m classes. Finally, the nearest neighbor interpolation method is used to interpolate the proximity points and complete the point cloud segmentation of the whole scene. Unlike the direct use of Pointnet, we obtain similar features between each piece, the selected regions of these features are different, and the obtained perceptual fields are also different. In the regions with a large number of hyperpoint families, the feature descriptions are more specific, and the opposite is true for the regions with a small number, we use this method to solve the different sizes of perceptual fields in different sparsity regions, and at the same time, the effective geometric feature set of ATSE is taken as the input of the network, and this high-dimensional input has better performance for the network and is more effective for learning. Then, the final segmentation result is obtained by the above interpolation formula.

3. Results and Discussion

3.1. 3D Radar Point Cloud Dataset. In this section, we evaluate the performance of the proposed method in the third part of 3D point cloud segmentation. We briefly describe the point cloud data used for the experiments and the implementation of the proposed method. Then, a brief description of the adopted evaluation indicators is given. Finally, we compared our classification results with relevant state-of-the-art classification methods.

(1) 3D radar point cloud dataset: The datasets can be divided into two groups according to the 3D lidar acquisition method and main applications (3D lidar data for semantic segmentation).

(2) Static datasets: scanners collect data from a static perspective, which is convenient for capturing static scenes such as a street view. Semantic3D [22] is currently the largest and most popular static dataset. Each frame is a single frame measured from a fixed position using a ground laser scanner. Ground, vegetation, and buildings were the main categories comprising this dataset, with fewer moving objects being compared. This three-dimensional semantic scene contains rural and urban scenes with three separate suburban categories. The per category proportions also vary widely. The large amount of data is beneficial for training deep learning models. Shown are the semantic3d dataset examples with point cloud data labels values from left to right in order of reflection intensity, RGB color, and category.

(3) Sequence data set: frame sequence point cloud data acquired from a mobile device. The algorithm was also tested on the dataset used by Mvf-CNN [23]. Scenario A contains a part of semantic3d and also the cloud of laser spots collected from a 3dslam backpack device, which is sparser. Scenario B, which has a data format of *X*, Y, *Z*, I, R, B includes bars, buildings, trees, road lights, traffic signs, cars, objects such as wires, towers, and pedestrians. Scenario C contains a lot of noise compared to B, and the color information and objects are incomplete unstructured point cloud data Scenario C has four classes of objects, car, tree, pedestrian, and building. Scenario *D* is an urban scene scanned using a TLS scanner and is derived from Roof.

(4) Bjfumap dataset: The high-precision point cloud data acquired by laser scanner can facilitate the 3D description of objects on the one hand, but at the same time, it needs to occupy a large storage space as well as requires a high processing speed. On the other hand, due to the increase in scanning accuracy, the point cloud density also increases dramatically, which brings great challenges to the processing of point cloud data. The point cloud data used in this paper was measured by the Special Equipment Research Center of Beijing Forestry University, where I work. In order to obtain this panoramic view, a total of 12 measurements were made in different areas and at different angles in front of the main building of Beijing Forestry University, and the measurement samples included spherical shrubs, trees, grass, step walls, buildings, and other targets. The multidimensional laser point cloud panorama of the forest environment information is shown in Figure 4, which is the result after opening with Geomagic software. You can see that after stitching the results of the 12 measurements together, there are still more than 19 million points composed in the whole point cloud data after the preliminary sampling process. What we are going to use is the forest understory resource environment other than buildings and roads, including purple-leaved bulbous shrubs, small green-leaved bulbous shrubs, large green-leaved bulbous shrubs, trees, buildings, and steps selected to have contact with plants as obstacles such as stones in the forest understory environment in a total of 4 categories.

We designed a pedestrian mobile backpack laser radar platform using 16-line laser radar and an IMU for fusion. The backpacks were recorded by walking in a closed-loop manner and scanning a campus environment in Beijing through Loam's slam construction method, containing construction, low vegetation, high vegetation, and four targets on the ground. In 4.2 b, an onboard laser radar slam system was set up using 16-line laser radar and two sick 511 laser radar for point cloud registration, and combined IMU to build a map through slam to collect point cloud data for construction, high vegetation, low vegetation, and shrub on campus.

A pedestrian mobile backpack laser radar platform is designed by using 16-line laser radar and an IMU for fusion. As shown in Figure 5, the backpacks were recorded by walking in a closed-loop manner and scanning a campus environment in Beijing through Loam's slam construction method, containing construction, low vegetation, high vegetation, and four targets on the ground. An onboard laser radar slam system was set up using 16-line laser radar and two sick 511 laser radars for point cloud registration and combine IMU to build a map through slam to collect point cloud data for construction, high vegetation, low vegetation, and shrub on campus.

In this subsection, three datasets used in the experiments are presented. Semantic 3D, a large outdoor point cloud dataset; Mvfcnn, a dataset used to validate the performance of our method on different hydrophobic point clouds and our self-built Bjfumap dataset.

3.2. Implementation and Evaluating Indicator. The detailed implementation of the algorithm is described as follows. The hyperparameters shown in the following table are selected, and a GPU containing a NVidia 1080ti dual graphics card is used to carry out the CUDA acceleration calculation and build the development environment of python3.7, pytorch1.0 on ubunta18.04. Parameters are set in Table 1. The point cloud has the property of rotational invariance, and the date is augmented by randomly rotating the point cloud around the *z*-axis before training. Dropout was also performed randomly on the cloud of partial points in the training set using the training method with random dropouts 0.3, 0.5, 0.7.

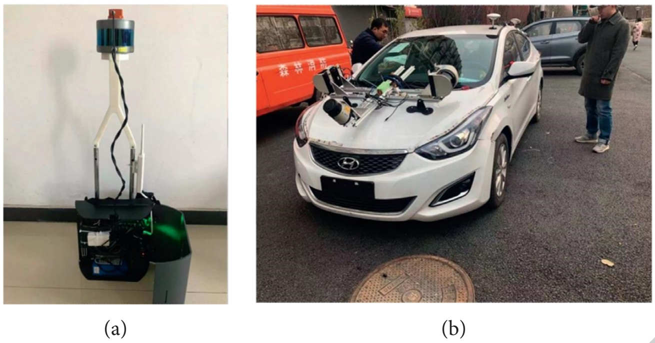


Figure 5: (a) backpack laser scanning equipment; (b) vehicle laser slam equipment.

Table 1: Parameter settings.

Training hyperparameters	Parameter values
Initial KNN parameter selection	10
Maximum number of iterative steps	500
Study attrition rate	0.65
Base study rate	0.002
Batch study size	2
Learning momentum	0.9
Block size	50

Performing random dropout on the data in every epoch trained can effectively improve the generalization of the training process, allowing the algorithm to perform well on a sparse point cloud.

$$IoU_i = \frac{c_i}{c_{ii} + \sum_{j\neq i} c_{ij} + \sum_{k\neq i} c_{ki}},$$

$$A\_IoU = \frac{\sum_{i=1}^{L} IoU_i}{L}, \tag{6}$$

$$OA = \frac{\sum_{i=1}^{L} c_{ii}}{\sum_{i=1}^{L}\sum_{k=1}^{L} c_{jk}}.$$

We followed the evaluation metrics of semantic3d, applying the recall IoU, each class of IoU, joint intersection, and total precision OA to evaluate the dataset used, where, c_{ij} is the number of samples predicted to be class j from the class i group structure. IoU is the evaluation index for each category, OA is the overall precision evaluation index of the dataset.

3.3. Segment Experiments and Analysis. In this subsection, the semantic3d dataset, mvfcnn dataset, and our bjfumap dataset are used for testing. First, the entry of our data into PointLAE is discussed, where we perform a hypersegmentation based on the aggregation features from the ATSE module. The results are shown below. After statistics of the three datasets are obtained, the semantic 3D dataset is used for training, as well as the mvfcnn dataset, where sparse point clouds were intermingled with dense point clouds, and our own acquired laser lidar dataset, bjfumap. PointLAE v2 was used in the semantic3d dataset and bjfumap dataset, and PointLAE v1 was used in the mvfcnn datasets (Table 2).

Table 2: Computation time (seconds) of different methods for semantic segmentation on semantic3D dataset.

Method	ATSE time	Over segmenttime	Train time	Total time
Pointnet2	6399	9755	52109	68263
PointLAE v1	7149	8409	22189	37747

The above table shows the training time statistics, and it can be seen that the total time, compared to pointnet2, the computation time of our network is significantly improved. We trained PointLAE on three datasets. The first row is the semantic3d dataset, the second row is the mvfcnn dataset, and the third row is our bjfumap dataset. The features proposed by the ATSE module were applied to the over-segmentation point cloud data, resulting in the results shown in Figure 6. Each color represents different over-segmentation classes and domain splits into more segments can be seen in geometrically complex regions. Similarly, sparse point clouds split into more segments. The segmentation effect also varies for each dataset.

We trained on the semantic3d dataset with PointLAE and the cross-entropy as adopted as the loss function. Semantic3d scenes were used for training, as shown in Figure 7, iterating over 500 epochs, and finally, the loss rate was fit to 0.21. The hardware and configuration mentioned in the previous subsection were used in the training process. As shown in Table 3, the training time was shortened compared to pointnet++. A visualization of the training effect is shown in Figure 8. The first line is the label visualization and the second line is our training result visualization. Precision evaluation of the training results, as shown in Table 3, we adopted the evaluation indices from the previous section to evaluate the training results. The overall precision OA reached 90.3 with a mIoU of 68.7, which performed well on targets such as high vegetation, low vegetation, and cars.

The model was tested on different sparse point clouds from the Mvfcnn dataset as shown in Figure 9, and the segmentation visualization is shown in Figure 10. The first row is the ground truth, and the second row shows the results of the segmentation labels. This dataset contains a point cloud from part of semantic3d. There are also sparse

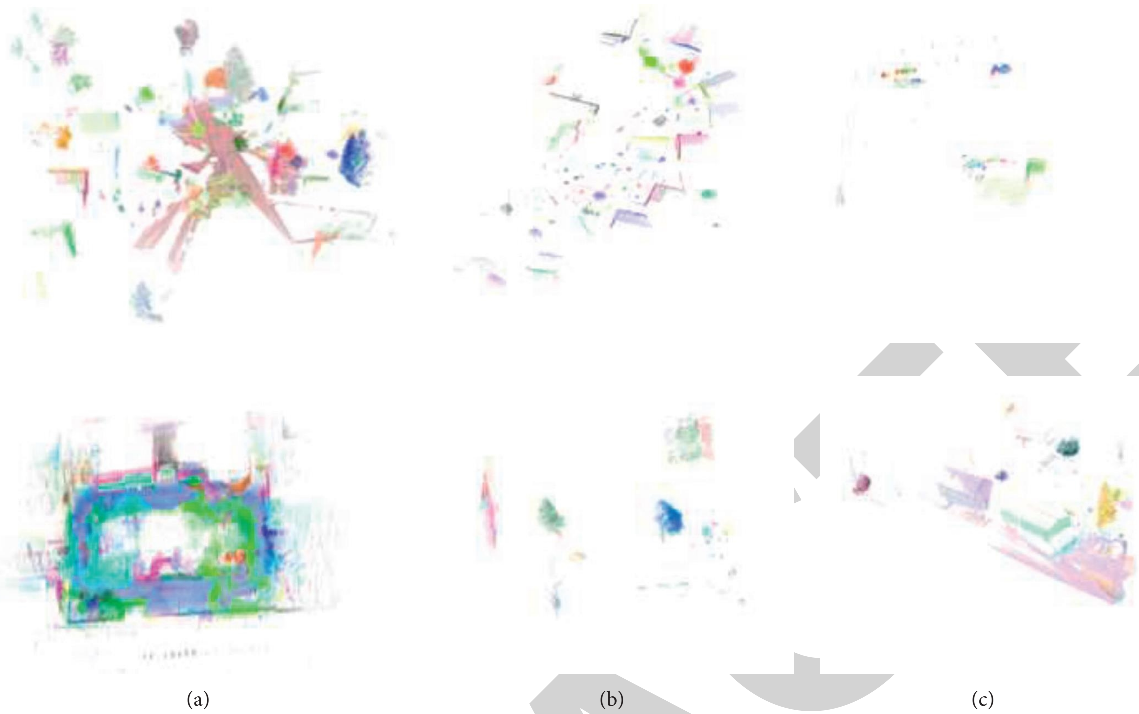


Figure 6: Example of PointLAE oversegmentation visualization on datasets, (a) is the semantic3d dataset, (b) is the mvfcnn dataset, and (c) is the Bjfumap dataset.

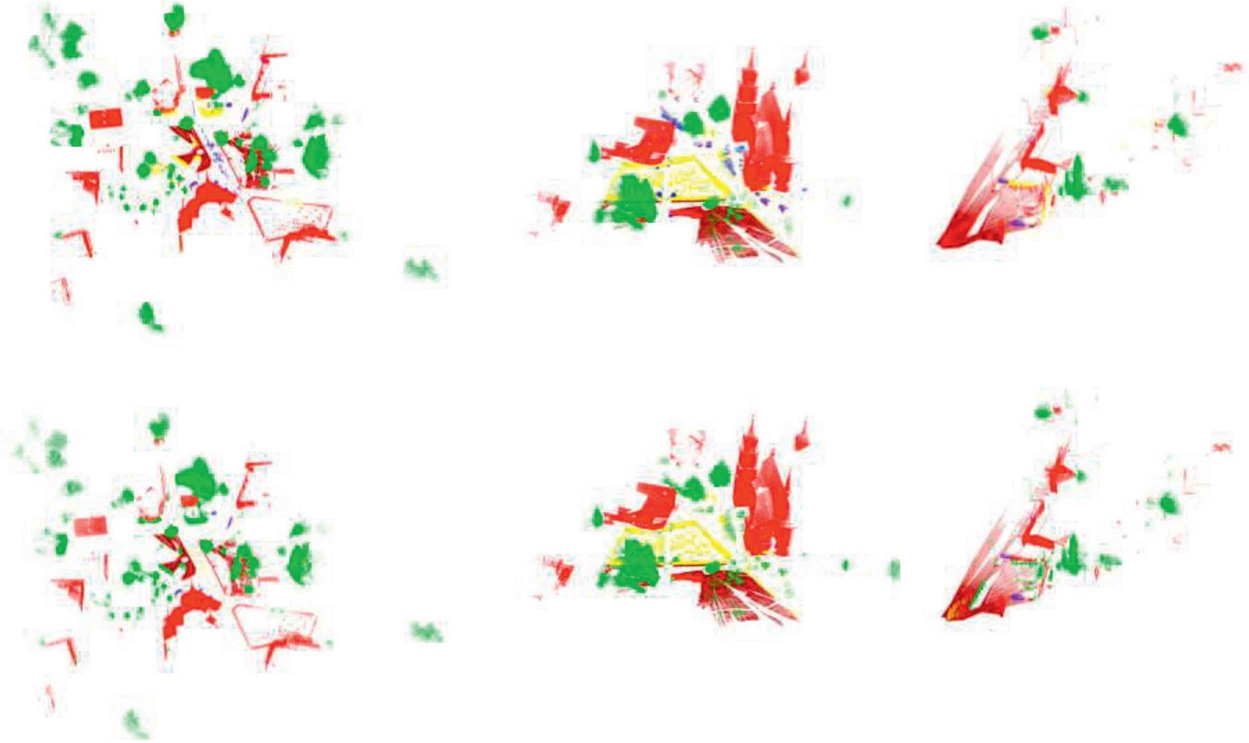
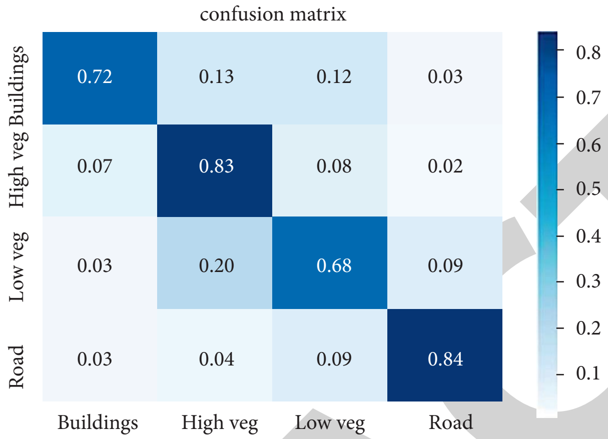


Figure 7: Example of PointLAE semantic segmentation visualization on the semantic3D dataset.

point clouds for which the original authors trained this dataset. Similarly, we trained the model in admixture and for which sparse scenes 3Performing the analysis. As shown in Table 4, contrasting Mvfcnn as well as 3DCNN [24] proposed by the original authors, our model effectively improved in overall accuracy, performing well in high vegetation, such as trees, and low vegetation, such as shrubs, with improved overall accuracy (Table 5).

Table 3: Intersection over union (IoU) metric and overall accuracy (OA) of different methods for different classes of the Semantic3D dataset.

Method	OA	mIoU	Road	High vegetation	Low vegetation	Buildings	Hard scape
Pointnet	82.5	52.1	74.5	59.8	60.8	81.7	33.2
Pointnet++	85.7	63.1	78.1	64.3	51.7	75.9	36.4
PointLAE v2	90.3	68.7	74.2	79.4	73.2	81.5	35.7


confusion matrix

	Buildings	High veg	Low veg	Road
Buildings	0.72	0.13	0.12	0.03
High veg	0.07	0.83	0.08	0.02
Low veg	0.03	0.20	0.68	0.09
Road	0.03	0.04	0.09	0.84

Figure 8: Confusion matrix of the experiment named pointLAE.

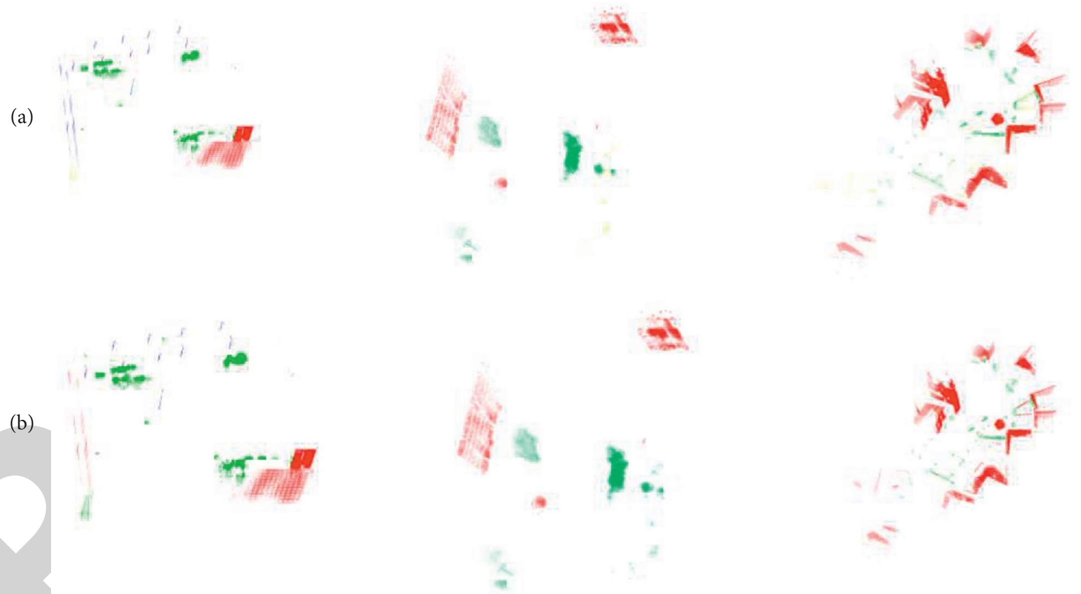


Figure 9: Examples of pointLAE semantic segmentation visualization on theMVFCNNdataset. (a) RGB point clouds; (b) prediction of semantic segmentation, and different colors represent different features.

We test this in our own extralarge scene point cloud registry acquired via the slam Bjfu dataset as shown in Figure 11, where on the left scenario 1 is the point cloud collected by backpack laser radars and on the right scenario 2 is the point cloud collected by vehicle laser radars, where the original point motions are reported without RGB information, and we label 4 of these classes, respectively, and trained using Pointlae v1 and Pointlae v2 against Pointnet + +, our network performed better. After increasing the reflection intensity as the input feature, our network PointLAE v2 performed better than Pointlae v1. Figure 8 is the visualization of the results of the training process. Table 5 reflects the overall accuracy evaluation of the method. Figure 8 is a confusion matrix to evaluate the correctness of classification.

3.4. Experiments and Analysis of ATSE Module. We calculated all the features mentioned in Section B of the semantic3d dataset and mvfcnn dataset, as well as our

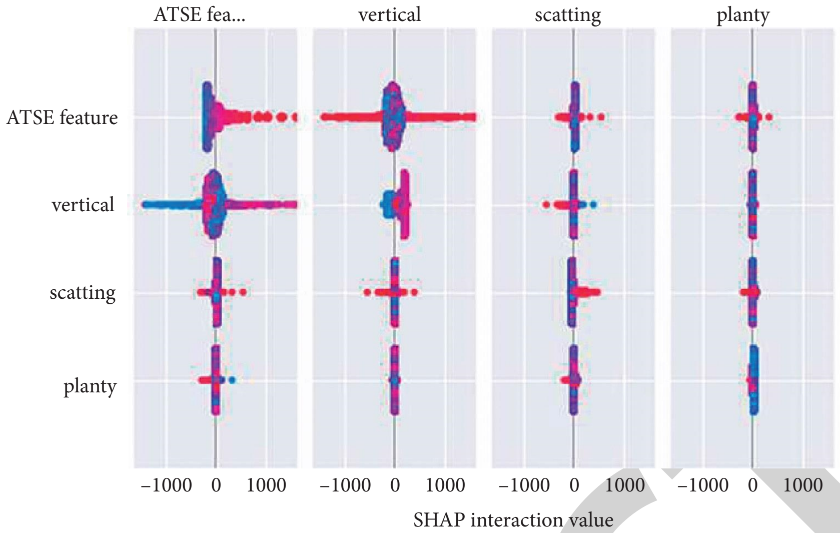


Figure 10: Feature interaction analysis.

Table 4: Intersection over union (IoU) metric and overall accuracy (OA) of different methods for different classes of the MVFCNN dataset.

Method	OA	mIoU	Manmade terrain	Natural terrain	Low vegetation
3DCNN	34.9	16.9	24.2	10.3	26.5
MVF CNN	85.1	72.9	75.4	70.3	77.4
PointLAE v1	86.2	73.75	65.9	71.2	78.5

Table 5: Intersection over union (IoU) metric and overall accuracy (OA) of different methods for different classes of the Semantic3D dataset.

PointLAE	OA	mIoU	Building	High vegetation	Low vegetation	Road
Pointnet++	40.8	32.65	37.1	40.2	16.7	36.6
PointLAE v1	76.7	66.47	69.9	69.7	52.7	73.6
PointLAE v2	77.4	73.75	73.2	71.7	55.5	68.8

bjfumap dataset, and based on the input of the ATSE v1 module, which does not contain reflection intensity into PointLAE, referred to as PointLAEV1, in our abbreviated PointLAEV2 with reflection intensity information ATSE v2 module. Using the bjfumap dataset as an example, we analyzed the validity of our ATSE model through shape [25]. As shown in Figure 12, plot the SHAP values for each of its point cloud features for each sample, which allows for a better understanding of the overall pattern and allows for the detection of predicted outliers. Each row represents a feature with the SHAP value in horizontal coordinates. A dot represents a sample and the color indicates the feature value (red high, blue low). Each row represents a feature and the abscissa is the shape value. This graph shows that the ATSE feature will have a positive effect on the prediction.

As shown in Figure 10, the interaction value is a way to generalize the SHAP value to higher-order interactions. The tree model implements a fast and accurate two-by-two interaction calculation, which returns a matrix for each prediction where the main effects are on the diagonal and the interaction effects are off the diagonal. These values often reveal interesting hidden relationships (interactions), which can also perform an analysis of the interaction of multiple variables to depict the variable versus target value under the interaction of two variables' impact. The red color indicates the greater value of the feature itself, the blue color indicates the smaller value of the feature itself, and at the same time, shap supports the analysis of a single sample. We selected the less recognizable low dwarf shrub.

As shown in Figure 13, the "explanation" above shows that each point cloud base feature has its own contribution to drive the model prediction from base value to final model output; the features that push the prediction up are shown in red, and the features that push the prediction down are shown in blue. Blue indicates that the contribution of the feature is negative, and red indicates the contribution of the feature. Literature is positive. The longest red bar is our ATSE feature, followed by the reflection intensity feature, which also demonstrates the importance of this feature of reflection intensity for the classification of the model. We concluded, through the above analysis, that the ATSE module can effectively aggregate effective features that have a large influence on the classification so that the ATSE features contribute more to the prediction.

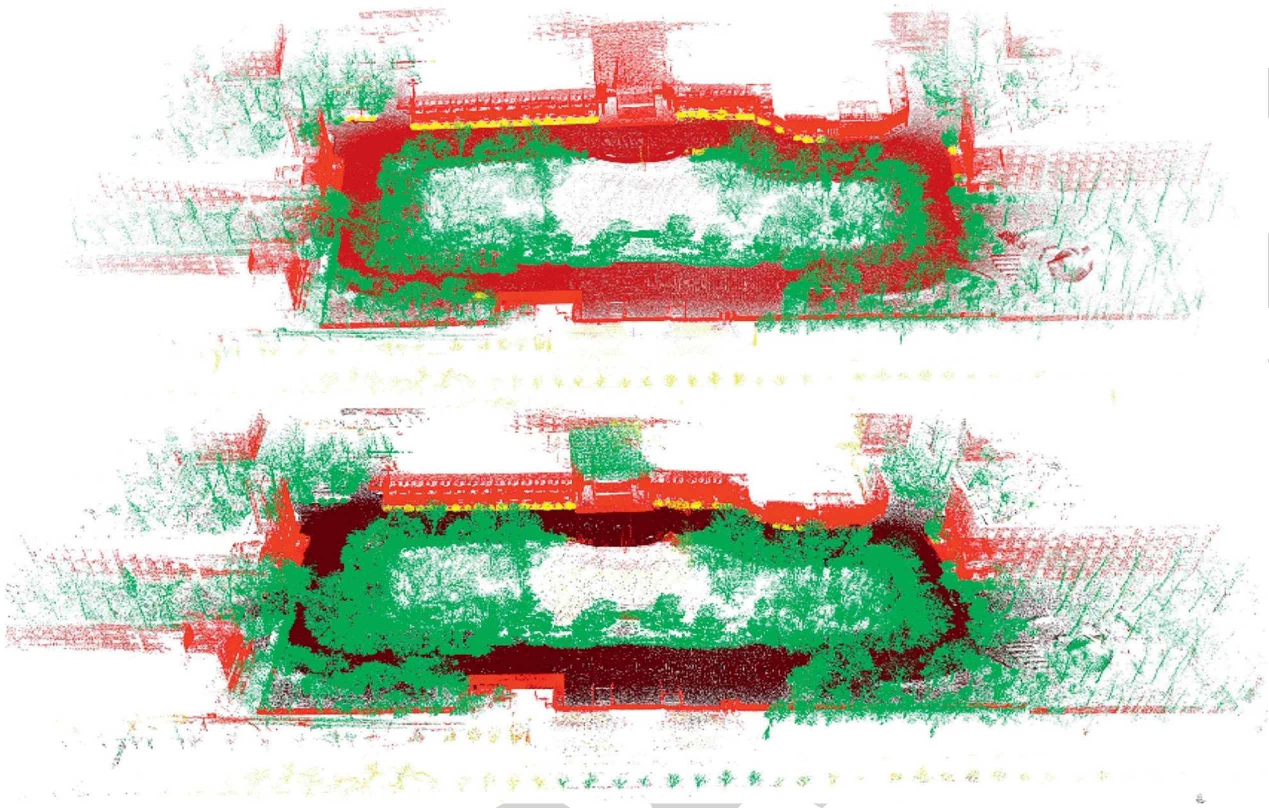


Figure 11: Examples of PointLAE semantic segmentation visualization on the Bjfumap datasets, sequentially for data acquisition by vehicle laser Liadars.

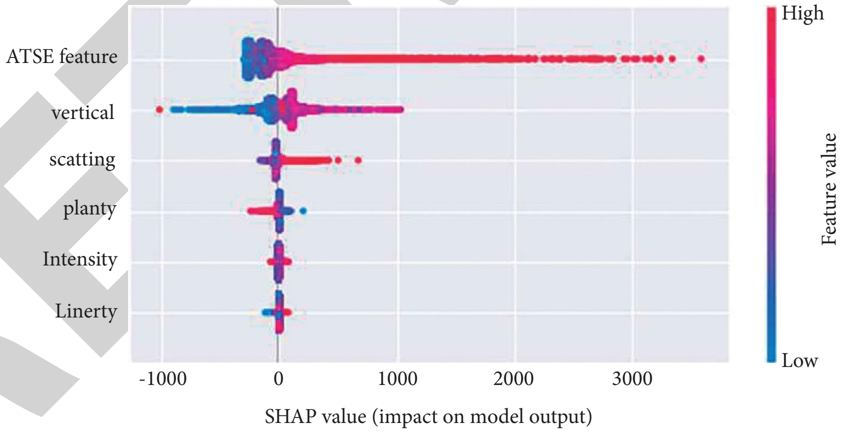


Figure 12: Feature ensemble analysis.

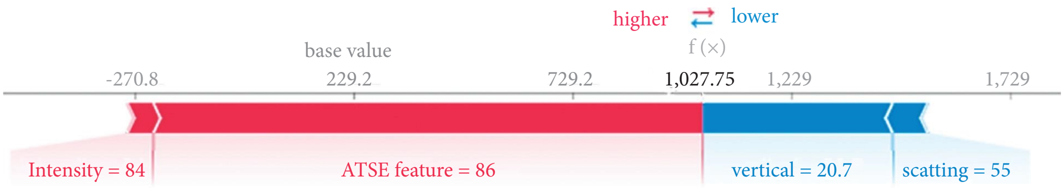


Figure 13: Feature contribution visualization.

4. Conclusion

In this paper, a pointLAE, a point cloud semantic segmentation network based on feature aggregation and energy optimization optimization, is proposed for large-scale 3D point clouds to deal with air crashes. It consists of two important modules. In the first module, multidimensional features of the point cloud are computed and optimized feature aggregation is selected to create the optimal features of the point cloud. In the second module, the point cloud is efficiently presegmented based on the optimized features and the optimal features are determined by an oversegmentation method based on energy functions. And training ah through a neural network. Finally, Pointnet and fully connected layers are used to obtain segmentation results. The experimental results for outdoor large-scale point cloud datasets (semantic3d, mvfcnn dataset, and Bjfumap) show that PointLAE can achieve better semantic segmentation results. Its total accuracy is 90.3% and 86.2%.77.4%, and miu is 68.7% and 86.2%,73.75. Comparing with other point cloud deep learning networks, our algorithm is suitable for complex and large scenes, and it is robust to sparse and uneven point clouds collected by lidar with different accuracies.

In the future, we should consider how to speed up the efficiency of the hyper-segmentation module and obtain better semantic segmentation results by optimizing the neural network structure. The proposed method will be transferred to other application fields such as agricultural image recognition, greenhouse environmental time-series prediction, food safety risk assessment, and image recognition, etc. [26–34].

Data Availability

The data used to support the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This paper was supported by the Research, Development and Demonstration of Key Technologies for Restoration of Fluid Fixed Vegetation in Countries along the Silk Road (No.2016YFE0203400-04), and National Key Technology R&D Program of China (No. 2021YFD2100605).

References

[1] Z. Zhirong Wu, S. Song, A. Khosla et al., “3d shapenets: a deep representation for volumetric shapes,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 12, p. 1912, 2015.

[2] D. Maturana and S. Scherer, “Voxnet: a 3D Convolutional neural network for real-time object recognition,” *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 13, p. 922, 2015.

[3] C. R. Qi et al., “Volumetric and multi-view cnns for object classification on 3d data,” *Computer Vision and Pattern Recognition*, p. 5648, 2016.

[4] Y. Li, S. Pirk, H. Su, C. Qi et al., “Fpnn: field probing neural networks for 3d data,” *Neural Information Processing Systems*, vol. 29, p. 307, 2016.

[5] D. Zeng Wang and I. Posner, “Voting for voting in online point cloud object detection,” *Robotics: Science and Systems XI*, vol. 1, p. 10, 2015.

[6] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multiview convolutional neural networks for 3d shape recognition,” *2015 IEEE International Conference on Computer Vision (ICCV)*, vol. 9, p. 945, 2015.

[7] J. Bruna, W. Zaremba, Szlam et al., “Spectral networks and locally connected networks on graphs,” *ArXiv preprint arXiv*, vol. 6, p. 1312, 2015.

[8] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, “Geodesic convolutional neural networks on riemannian manifolds,” *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, vol. 8, p. 3745, 2015.

[9] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “Pointnet: deep learning on point sets for 3d classification and segmentation,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, p. 652, 2017.

[10] C. R. QiL. Yi et al., “Pointnet++: deep hierarchical feature learning on point sets in a metric space,” *ArXiv preprint arXiv*, vol. 1, p. 1706, 2017.

[11] M. Jiang, Y. Wu, T. Zhao, Zhao et al., “Pointsift: a sift-like network module for 3d point cloud semantic segmentation,” *ArXiv preprint arXiv*, vol. vol.2, p. 1807, 2018.

[12] Y. Li, R. Bu, M. Sun, Wu et al., “Pointcnn: convolution on x-transformed points,” *Advances in Neural Information Processing Systems*, vol. 31, p. 820, 2018.

[13] L. Landrieu and M. Simonovsky, “Large-scale point cloud semantic segmentation with superpoint graphs,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, vol. 34, pp. 4558–4567, 2018.

[14] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Pollefeys, and M. au, “Semantic3D.net: a new large-scale point cloud classification benchmark,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1/W1, pp. 91–98, 2017.

[15] V. Sakhre, U. P. Singh, and S. Jain, “FCPN approach for uncertain nonlinear dynamical system with unknown disturbance,” *International Journal of Fuzzy Systems*, vol. 19, no. 2, pp. 452–469, 2017.

[16] Y. Zhang, X. Chen, D. Guo, M. Song, and Y. X. Teng, “PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows,” *IEEE Access*, vol. 7, pp. 119904–119916, 2019.

[17] H. Thomas, C. R. Qi, J.-E. Deschaud, and B. F. L. Marcotegui, “Kpconv: flexible and deformable convolution for point clouds,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 23, pp. 6411–6420, 2019.

[18] M. Weinmann, S. Urban, S. B. Jutzi, and C. au, “Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas,” *Computers & Graphics*, vol. 49, pp. 47–57, 2015.

[19] T. Chen and C. Guestrin, “XGBoost,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 22, p. 785, 2016.

[20] S. Guinard et al., “Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds,” *ISPRS Workshop*, vol. 3, p. 1321, 2018.

[21] L. Obozinski and G. au, “Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs,” *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1724–1766, 2017.

[22] Y. Li, G. Tong, X. Li, L. Peng, and H. au, “MVF-CNN: fusion of multilevel features for large-scale point cloud classification,” *IEEE Access*, vol. 7, pp. 46522–46537, 2019.

[23] S. M. Lundberg et al., “A unified approach to interpreting model predictions,” *Neural information processing systems*, vol. 31, p. 4768, 2017.

[24] X. Cao and W. au, “Non-iterative approaches in training feed-forward neural networks and their applications,” *Soft Computing*, vol. 22, no. 11, pp. 3473–3476, 2018.

[25] L. Zhang, G. Zhu, P. Shen, and J. Song, “Learning satiotemporal features using 3dcnn and convolutional lstm for gesture recognition,” *Computer Vision Workshops*, vol. 1, p. 3120, 2017.

[26] X.-B. Jin, W.-Z. Zheng, and J.-L. X.-Y. M. Q.-C. S. Kong, “Deep-learning temporal predictor via bidirectional self-attentive encoder-decoder framework for IOT-based environmental sensing in intelligent greenhouse,” *Agriculture*, vol. 11, no. 8, p. 802, 2021.

[27] X. B. Jin, W. Z. Zheng, J. L. Kong, and X.-Y. Y.-T. T.-L. S. Wang, “Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization,” *Energies*, vol. 14, no. 6, pp. 1596–1614, 2021.

[28] J. Kong, C. Yang, J. Wang, and X. M. X. S. Wang, “Deep-stacking network approach by multisource data mining for hazardous risk identification in IoT-based intelligent food management systems,” *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–16, 2021.

[29] J. Kong, H. Wang, X. Wang, and X. X. S. Jin, “Multi-stream hybrid architecture based on cross-level fusion strategy for fine-grained crop species recognition in precision agriculture,” *Computers and Electronics in Agriculture*, vol. 185, p. 106134, 2021.

[30] X. B. Jin, W.-T. Gong, J. L. Kong, and Y.-T. T.-L. Bai, “PFVAE: a planar flow-based variational auto-encoder prediction model for time series data,” *Mathematics*, vol. 10, no. 4, p. 610, 2022.

[31] X. Jin, J. Zhang, J. Kong, and T. Y. Su, “A reversible automatic selection normalization (RASN) deep network for predicting in the smart agriculture system,” *Agronomy*, vol. 12, no. 3, p. 591, 2022.

[32] X.-B. Jin, W.-T. Gong, J. L. Kong, and Y.-T. T.-L. Bai, “A variational bayesian deep network with data self-screening layer for massive time-series data forecasting,” *Entropy*, vol. 24, no. 3, p. 335, 2022.

[33] J. Kong, H. Wang, C. Yang, and M. X. Jin, “A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition,” *Agriculture*, vol. 12, no. 4, 500 pages, 2022.

[34] Y. Y. Zheng, J. L. Kong, X. B. Jin, and X.-Y. M. Wang, “CropDeep: the crop vision dataset for deep-learning-based classification and detection in precision agriculture,” *Sensors*, vol. 19, no. 5, p. 1058, 2019.