ISPRS Journal of Photogrammetry and Remote Sensing 193 (2022) 115–136

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

# A joint deep learning network of point clouds and multiple views for roadside object classification from lidar point clouds

Lina Fang [a], Zhilong You [a], Guixi Shen [a], Yiping Chen [b,*], Jianrong Li [c]

[a] Academy of Digital China (Fujian), Fuzhou University, Fuzhou 350108, China
[b] School of Geospatial Engineering and Science, Sun Yat-sen University, 519082 Zhuhai, China
[c] Fuzhou Investigation & Surveying Institute Co. Ltd, Fuzhou 350108, China

ARTICLE INFO

*Keywords:*
Mobile laser scanning systems
Point cloud classification
Multiview images
Deep learning
Attention mechanism

ABSTRACT

Urban management and survey departments have begun investigating the feasibility of acquiring data from various laser scanning systems for urban infrastructure measurements and assessments. Roadside objects such as cars, trees, traffic poles, pedestrians, bicycles and e-bicycles describe the static and dynamic urban information available for acquisition. Because of the unstructured nature of 3D point clouds, the rich targets in complex road scenes, and the varying scales of roadside objects, finely classifying these roadside objects from various point clouds is a challenging task. In this paper, we integrate two representations of roadside objects, point clouds and multiview images to propose a point-group-view network named PGVNet for classifying roadside objects into cars, trees, traffic poles, and small objects (pedestrians, bicycles and e-bicycles) from generalized point clouds. To utilize the topological information of the point clouds, we propose a graph attention convolution operation called AtEdgeConv to mine the relationship among the local points and to extract local geometric features. In addition, we employ a hierarchical view-group-object architecture to diminish the redundant information between similar views and to obtain salient viewwise global features. To fuse the local geometric features from the point clouds and the global features from multiview images, we stack an attention-guided fusion network in PGVNet. In particular, we quantify and leverage the global features as an attention mask to capture the intrinsic correlation and discriminability of the local geometric features, which contributes to recognizing the different roadside objects with similar shapes. To verify the effectiveness and generalization of our methods, we conduct extensive experiments on six test datasets of different urban scenes, which were captured by different laser scanning systems, including mobile laser scanning (MLS) systems, unmanned aerial vehicle (UAV)-based laser scanning (ULS) systems and backpack laser scanning (BLS) systems. Experimental results, and comparisons with state-of-the-art methods, demonstrate that the PGVNet model is able to effectively identify various cars, trees, traffic poles and small objects from generalized point clouds, and achieves promising performances on roadside object classifications, with an overall accuracy of 95.76%. Our code is released on https://github.com/flidarcode/PGVNet.

## 1. Introduction

Roadside cars, trees, traffic poles (street lights, traffic signs) and small objects (pedestrians, bicycles and e-bicycles) are important urban facilities and information targets. Their position, category and semantic information are the core elements used to describe the static and dynamic urban environment, which plays an important role in numerous applications, such as intelligent transportation, navigation and position service, automatic driving and high-definition maps. As an advanced 3D surveying and mapping technology, laser scanning systems such as mobile laser scanning systems (MLSs) are becoming increasingly available and affordable for acquiring the high-precision spatial information of road scenes in the format of point clouds. Point clouds preserve raw geometric information with 3D coordinates and have three core characteristics, including being disordered, interactivity among the points, and invariance under transformation, which allow a better understanding of roadside objects and the corresponding environments. However, roadside object classification from point clouds still faces several significant challenges due to the rich targets in complex road scenes, uneven point density, occlusion, the diversity of types, the

* Corresponding author.
*E-mail address:* chenyiping@xmu.edu.cn (Y. Chen).

https://doi.org/10.1016/j.isprsjprs.2022.08.022
Received 5 March 2022; Received in revised form 19 August 2022; Accepted 23 August 2022
Available online 22 September 2022

varying scales of roadside objects, and the unstructured nature of 3D point clouds (Yang et al., 2017; Che et al., 2019; Guo et al., 2020). On this basis, this paper focuses on an analysis of the deep learning methods on point clouds for processing roadside object classification.

Some existing studies deal with roadside object classification as a generic classification problem, like classifying all objects in the data rather than a specific class (Xiao et al., 2016; Dong et al., 2017). One common procedure is to remove points belonging to the ground and building façades, and then cluster the remaining points into individual objects (Yan et al., 2017). This process is predominantly employed as a pre-processing step in roadside classification and can significantly promotes efficiency gains in large-scale scenes. Besides, it provides a more compact representation of roadside objects at the object level, instead of the individual point level (Poux et al., 2022). Machine learning for roadside object classification is popular for 3D point clouds. Some existing methods are endeavored to describe handcraft geometric, contextual, intensity and color features of roadside objects and fed them into the unsupervised or supervised classifiers (Lehtomäki et al., 2015; Yang et al., 2017). The challenge is that handcraft features can be affected by many factors, e.g., scanning perspective, point density, noise and occlusion (Mi et al., 2021a). In addition, shallow learning classifiers (SVM, Radom Forest, etc.) are limited by modeling capacity and need full and discriminative features to achieve good performance (Poux and Billen, 2019).

Since deep learning has achieved overwhelming success in capturing high-level distinguishing features, an increasing number of studies have explored deep learning frameworks to address roadside object classification tasks (Mi et al., 2021b; Han et al., 2021). Existing deep learning methods on 3D roadside object classification are roughly grouped into volumetric-based, multiview-based, point-based methods (Guo et al., 2020) in terms of the representation of the input for neural networks (Li et al., 2021). At an early stage, many studies converted the irregular point clouds of roadside objects into regular 3D grids and implemented the 3D convolutional neural networks (CNNs) to classify them. Because multiple adjacent points are quantized into a single grid, the performance of volumetric-based methods heavily relies on the voxelization resolution. Hence, volumetric-based methods need to balance the data sparsity and computational complexity when addressing roadside objects with various sizes and scales. To represent the point cloud consistently, multiview-based methods usually generate a series of views from different perspectives. It is natural and easy to exploit well-established 2D convolutional neural networks (CNNs), such as VGG (Simonyan and Zisserman, 2014), ResNet (He et al., 2016), and AlexNet (Krizhevsky et al., 2012), with multiview data and capture the viewwise features. Although multiview-based methods show great advantages in efficiently extracting advanced global features and are very hardware friendly, these methods treat all views equally and may ignore the contextual relationships between a group of views, resulting in the loss of local detailed information. Recently, some deep learning networks have been directly employed on raw point clouds to better maintain 3D geometric information. As a pioneering work, PointNet (Qi et al., 2017a) extracted the spatial feature of each point independently and then aggregated these pointwise features into global features by a symmetric function, which was used to address the problem of point cloud irregularity and disorder. Meanwhile, graph convolutional neural networks have attracted much attention since 2018 because they can normalize and order point clouds (Defferrard et al., 2016; Zhang and Rabbat, 2018; Shi and Rajkumar, 2020). Since PointNet cannot learn the relationship between points, subsequent models, including PointNet++ (Qi et al., 2017b), DGCNN (Wang et al., 2019b) and PointCNN (Shi and Rajkumar, 2020), were implemented on a local neighborhood of points instead of on the independent points and achieved better performance on 3D object recognition. In fact, different local patches play different roles in roadside object recognition when treating the same type of objects with varying sizes and incompleteness. These methods often enlarge the receptive field or stack various multiscale models to address this problem, which results in huge computation costs. Therefore, the key challenge for point-based methods is to efficiently explore the relationships between different local structures when processing large-scale outdoor point clouds.

Recently, some studies integrated multimodal deep learning models with different representations as input and achieved state-of-the-art performance. Among these works, point-based and multiview-based 3D shape representations have recently attracted more attention, and their corresponding deep models have achieved promising performance on 3D shape recognition from public datasets such as ModelNet 40 (Vishwanath et al., 2009). However, there is little research that jointly considers point cloud data and multiview image data for roadside object classification from outdoor point clouds, which is, in our consideration, beneficial and complement each other. Motivated by the merits and limitations of multiview-based and point-based methods, we propose a jointed point-group-views deep learning framework called PGVNet on mixed representations, including multiview images and point clouds, to classify roadside objects. The main research contributions of the proposed methods can be summarized as follows:

First, we jointly consider point cloud data and multiview image data for roadside object classification and propose a novel point-group-view network to classify large-scale scene roadside objects, which achieves state-of-the-art results as compared with eight existing methods.

Second, a hierarchical view-group-object architecture is presented to reduce the redundant information between similar views, and it splits all the views into different groups to reduce the cost in the multiview branch of our PGVNet.

Third, an attention-guided fusion module is adopted to aggregate the global features from the multiview images and the local geometric features from the point clouds and obtain the intrinsic correlation and discriminability of the local geometric features, which significantly improves the classification performance on six large-scale scenes.

Finally, we prove that our PGVNet trained by MLS datasets is capable of a good generalization classification performance by testing it on high-density ULS point cloud data and BLS point cloud data. The corresponding results reached 90.74 % OA and 99.16 % OA, respectively.

The rest of this paper is structured as follows: we provide an overview of the recent studies on implementing deep learning on roadside object classification from MLS point clouds and 3D shape recognition in Section 2. Section 3 details our PGVNet model. Section 4 presents the experimental results and relevant discussions. Section 5 concludes the study with a summary of the findings.

## 2. Related work

In recent years, following the great success of deep learning on 2D images, many methods that apply deep models to 3D object recognition have achieved good performance. In terms of the representation of 3D objects, deep learning methods can be roughly divided into volumetric-based, multiview-based, point-based and multimodal fusion methods.

### 2.1. Volumetric-based methods

Most CNNs are designed for regular grids or images and cannot be directly implemented on an unordered point cloud. Hence, at an early stage, studies converted the raw point clouds into regular voxelized data and employed CNNs on the voxels to recognize 3D objects. Wu et al. (2015) utilized a binary voxel to represent 3D objects and proposed ShapeNet, which explored a convolutional deep belief network to learn the geometric features of complex 3D shapes. Maturana and Scherer (2015) presented a volumetric-based deep learning framework named Voxnet, which transformed point clouds into voxel grids and then built a 3D CNN to predict the semantic labels of grid data. Similarly, Brock et al. (2016) employed a neural network for the voxel representation of 3D objects and achieved good classification performance. Although these methods easily learned the embedding of grids, they were difficult to

scale well to outdoor 3D point clouds because the runtime and memory costs increase cubically with resolution increase. Hence, some hierarchical and compact structures, such as octree and $k_d$-tree, were explored to decrease the computation and memory costs (Klokov and Lempitsky, 2017; Riegler et al., 2017). Riegler et al. (2017) proposed a deep learning network on sparse 3D data called OctNet, which voxelizes point clouds into the hybrid grid-octree structure. Benefiting from an efficient octree structure, OctNet enabled 3D convolutional networks on desired voxels by dynamically allocating memory and computation to the relevant dense regions. In addition, some studies partitioned the raw point clouds into multiscale voxel grids and employed the 3D fully convolutional neural networks to overcome the problem of information loss (Roynard et al., 2018a; Ye et al., 2020). Although these methods have achieved good classification results, they also face problems such as heavy memory consumption and high computational costs.

### 2.2. Multiview-based methods

The main idea of the multiview-based methods is to utilize a set of views captured from different viewpoints to represent a 3D object and then implement image-based deep neural networks to learn object features from the views. Inspired by 2D convolutional neural networks, Su et al. (2015) proposed a multiview convolutional neural network (MVCNN) to address the problem of 3D object recognition. MVCNN successfully applied a group of shared weight CNNs to learn advanced features from the multiview images and encouraged the application of existing deep learning frameworks to the point cloud domain. The key challenge of multiview-based methods is to aggregate viewwise features into a discriminative global descriptor. MVCNN leverages max pooling layers to address this issue, which preserves the maximum values from a certain view. Because max pooling layer may lose some key information, Feng et al. (2018) proposed a group-view convolutional neural network named GVCNN to generate discriminative viewwise descriptions. To obtain a more compact global descriptor, a grouping module is used to exploit the view-view relationship over a group of views and assign different weights to different groups. Compared with MVCNN, GVCNN can capture more distinguishable information and achieve better performance on 3D shape recognition. In addition, Yang and Wang (2019) introduced a relation network to effectively encode the region-to-region and view-to-view relationships between different view images and then aggerated the viewwise features into a discriminative 3D object representation. Similarly, Xu et al. (2021) proposed a correspondence-aware deep learning network (CAR-Net) to exploit the intra-view and cross-view relationships. To capture significant geometric cues for 3D shape recognition, a correspondence-aware representation (CAR) module was stacked in CAR-Net to encode the spatial relations of views according to the spatial arrangement of the different object parts and the symmetry of the object. However, additional views would introduce more redundant information and decrease the discriminative information. Hence, Luo et al. (2019) only generated three feature descriptors from three different perspectives and implemented two fusion networks to combine the multiview features into the high-level features of the MLS point cloud. Fang et al. (2020) utilized three binary images to represent roadside objects and employed a deep belief network to effectively classify independent traffic objects derived from MLS point clouds. Recently, inspired by the attention mechanism, Nie et al. (2021) introduced a deep-attention network (DAN) for 3D shape representation. The self-attention mechanism was introduced to reduce the redundant information and highlight the distinctive views by weighting the correlations among the views. After reviewing the above methods, we found that these view-based methods consider two problems, multiview information combinations and the removal of redundancy information, separately. These two problems are closely related and should be addressed simultaneously.

### 2.3. Point-based methods

Point-based deep learning methods have received much attention since PointNet (Qi et al., 2017a) was proposed in 2017. These methods performed deep learning directly on irregular and disordered 3D point clouds (Boulch et al., 2020; Nie et al., 2022). PointNet applied some shared multilayer perceptrons (MLPs) to learn the pointwise features and then used max-pooling layer as the symmetric aggregation function to generate the global features. Because each point are adressed independently, PointNet is unable to capture the fine-grained geometric information between points. Based on PointNet, Qi et al. (2017b) introduced a hierarchical network called PointNet++ to hierarchically and progressively aggregate the fine-gained geometric information from the local point sets. In particular, multiscale grouping and multiresolution grouping algorithms are also introduced in PointNet++ to eliminate the negative effects of the nonuniformity and uneven density of the point clouds. In addition, some studies implemented 3D convolution kernels on neighboring points to learn the local geometric features (Thomas et al., 2018; Varney and Asari, 2022). In PointCNN (Li et al., 2018), an X-Conv operator was used to permute and weight the input points and features. Then, PointCNN acquired local information from the neighboring points through a traditional convolution operation, which can be interpreted as a weighted sum operation over the local point sets. Inspired by the convolutional neural networks in 2D images, Wu et al. (2019) proposed a density re-weighted convolution named pointConv, which has the capability of implementing 3D continuous convolution on any set of 3D points. Thomas et al. (2019) designed a flexible and deformable kernel point convolution named KPConv to explore the correlation between the point clouds. In contrast to a fixed grid convolution, kernel point convolution is more flexible for interpolating point features. Further, Choy et al. (2019) proposed a set of generalized sparse convolutions and implemented them in an auto-differentiation library MinkowskiEngine. As a high-level extension of the traditional 2D convolution, sparse convolutions only compute the outputs for predefined coordinates and store the results into sparse tensors, which significantly reduce the computation cost of point clouds segmentation and classification.

To utilize the topological information of point clouds, Wang et al. (2019b) proposed a dynamic graph convolution operation (EdgeConv), which regards points as nodes of a graph and learns the edge features associated with the neighbors of each point in the spatial domain. By applying a convolution operation to the *k*-nearest neighbor graphs, the EdgeConv operation dynamically captures geometric information from the point clouds. To preserve the direction relationship between the points, the GeoConv layer in Geo-CNN (Lan et al., 2019) was applied to decompose the edge features into different directions and then aggregated the features along each direction. Zhao et al. (2019) proposed an adaptive feature adjustment module to aggregate local features by learning the interaction between the points and showed that an improved aggregate method could facilitate feature expressiveness. Although these methods improved the way of aggregating local features, they mainly chose customizable spatial filters locally applied on a regular receptive field (Thomas et al., 2019; Liu et al., 2022). The weights of the convolution operator are often determined by the geomatric distance between points and are fixed at specific positions. Recently, some work pursued the idea of using the attention mechanism to learn more adaptive local features. Chen et al. (2019) developed a two-stage object detection approach that combined the raw point cloud coordinate and the points feature with an attention mechanism to obtain more accurate localization and contextual information. Wang et al. (2019a) designed a graph attention convolution for a more flexible local feature extraction, in which the convolution kernel is determined by the learnable attentional weights. Similarly, Wen et al. (2021) combined the attention mechanism with a graph convolution to extract the global and local features to adress airborne lidar point cloud classification. However, limited by the convolution size and perception field, most of the existing

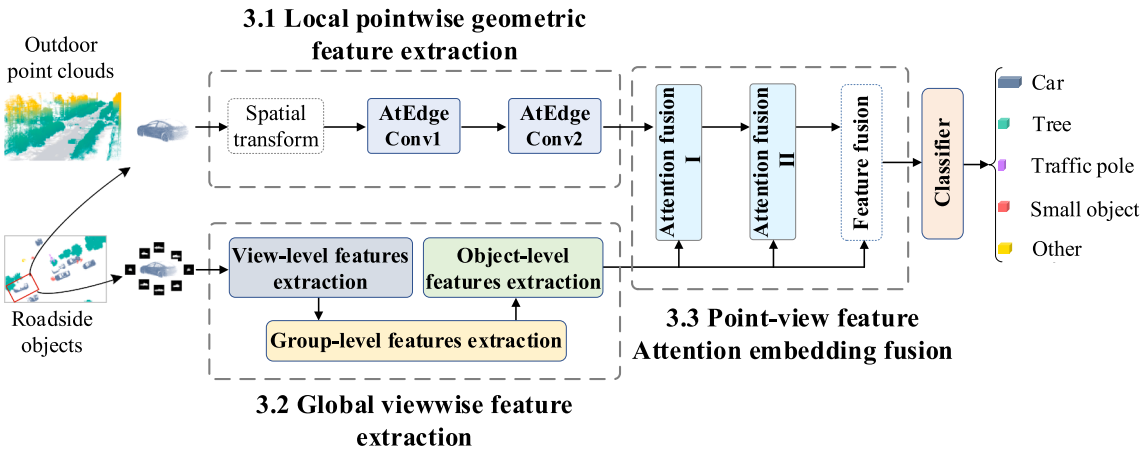



**Fig. 1.** Architecture of our PGVNet for roadside object classification. Our input is two modalities of a roadside object: point clouds and 2D views. Our PGVNet is composed of 3 parts: local pointwise geometric feature extraction, global viewwise feature extraction and point-view feature embedding attention fusion. The final feature after fusion is used for roadside object classification.

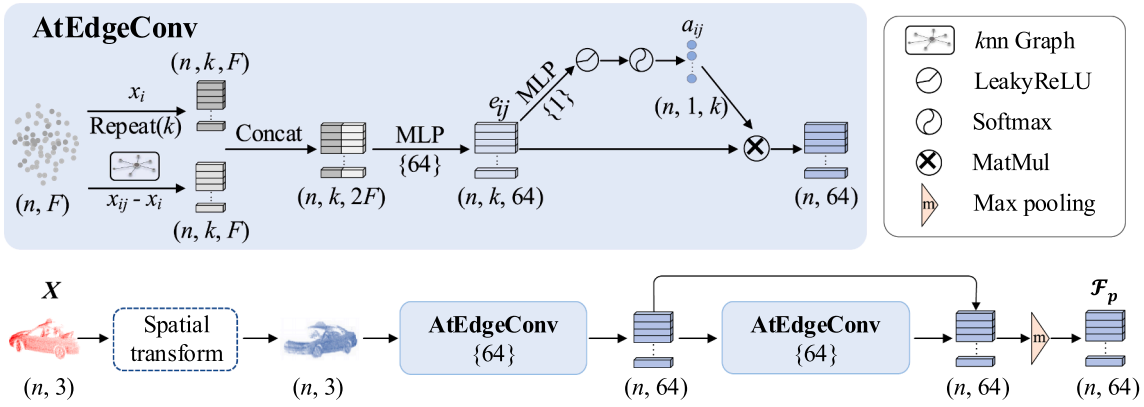



**Fig. 2.** Local geometric feature extraction network. The top is the AtEdgeConv layer, which captures the local geometric features of *k*nn graph structures for the point clouds. The bottom is the framework of the local geometric feature extraction network, which stacks two AtEdgeConv layers to explore the discriminative pointwise geometric features.

point-based methods excelle at capturing local geometric feature but are hard to mine the correlationship between them, which limites the performance and efficiency when dealing with large-scale scenarios with uneven target proportions and a sparse dispersion of points.

### 2.4. Multimodal fusion methods

Because the different representations of the 3D objects describe different characteristics for object recognition, fusing different models is reasonable and beneficial if used in an effective manner. In autonomous driving filed, multisensor data such as images and point clouds are often fused for 3D object detection and achieve good performance (Guo et al., 2016; Meyer et al., 2019). Xu et al. (2018) proposed PointFusion, a fusion network, to combine image data and point cloud data. A CNN and a PointNet are used to address the image data and point cloud data, respectively. Then, the outputs of the two branch architectures are fused by a deep sensor fusion network to estimate the 3D object bounding boxes. To obtain more discriminate features, You et al. (2018) introduced a point-view network named PVNet, which embeds an attention fusion network to combine the outputs of the multiview branch and point cloud branch. Specifically, global features from multiview images are used as soft attention masks to generate the attention-aware features of the point cloud models, which describe the significance of the different local structures. Moreover, You et al. (2019) exploited two kinds of relations, the point-multiview relation and point-single-view relation, to combine the pointwise features and viewwise features together and attained a remarkable performance on ModelNet40 (Vishwanath et al., 2009). Due to camera angles, a single view only preserves the partial local structures of an entire 3D shape, but more view features may repress discriminative features for shape recognition. In addition, these multimodal fusion methods treat multiview features equally when fused with point cloud features, which ignores some important relationships between the point cloud data and multiview data.

## 3. The joint network: PGVNet

To recognize roadside objects from outdoor point clouds, we first implement a coarse-to-fine segmentation method by Fang et al. (2020) to extract individual roadside objects from non-ground points, which cluster non-ground points using geometric distance, and further split overlapped objects into individual objects based on the Ncut algorithm. Then we design a joint point-group-view network called PGVNet to fuse the global features, represented by the multiview and local features described by the point clouds, into a discriminative descriptor for each roadside object. As illustrated in Fig. 1, we stack three blocks in PGVNet as follows:

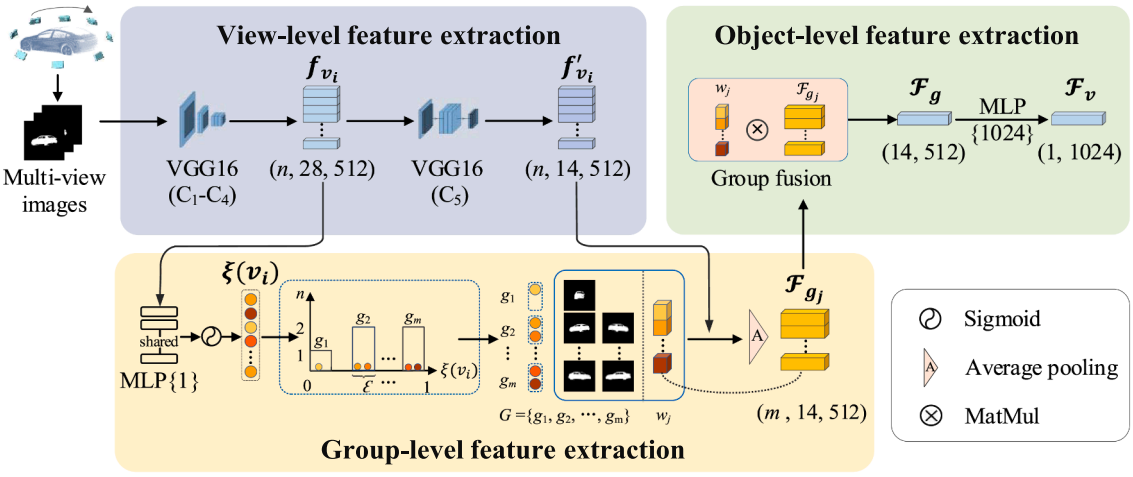



**Fig. 3.** Global viewwise feature extraction network. We first employ a pretrained VGG 16 to extract view-level features. We then group the views and measure the weight of each group, which guides further group fusion. After fusing intergroup view features, we reduce redundant information between views and explore the group-level features, which is a more compact representation of views. After aggregating view-level and group-level features, the final object-level descriptor of views is obtained.

1) Local pointwise geometric feature extraction: Capturing local distinctive features of the point clouds by embedding a self-attention mechanism in the DGCNN framework;
2) Global viewwise feature extraction: Integrating the view-level, group-level and shape-level features into a global viewwise descriptor.
3) Point-view feature fusion: Two attention fusion layers are used to fully leverage the global viewwise features and local pointwise features.

In this way, the local features from point clouds and global features from multiview images can be combined into a more distinguishing descriptor. We then fed it into the classifier to recognize the cars, trees, traffic poles (street lights, traffic signs) and small objects (pedestrians, bicycles and e-bicycles). Finally, our network outputs the probability value of the objects assigned to $C$ classes.

### 3.1. Local pointwise geometric feature extraction

Due to the complex road environment, various types and vary-level incompleteness, it is important to explore the discriminative fine-grained geometric features from the point clouds. To this end, we define a graph attention edge convolution (AtEdgeConv) layer, as shown in Fig. 2. The AtEdgeConv layer takes $n$ points with $F$ dimensions of individual roadside objects as input, denoted as $X = \{x_i \in \mathbb{R}^F, i = 1,2,\ldots, n\}$, where $x_i$ is the feature vector of point $i$. Instead of capturing point features directly from their embedding, AtEdgeConv aims to dynamically aggregate the local graph information represented by edge features $e_{ij}$, which combines the geometric relationship between the node $x_i$ and the local neighbour $x_{ij}$.

$$e_{ij} = h_\Theta\left(\left[x_i || x_{ij} - x_i\right]\right) \tag{1}$$

where $h_\Theta$ denotes the fusion function and $\Theta$ is the corresponding learned parameter. To measure the importance of the neighbors, we compute the normalized attention coefficients $\alpha_{ij}$ depending on the corresponding edge features $e_{ij}$. Then, we utilize the normalized coefficients $\alpha_{ij}$ as graph attention marks and aggregate them with edge features $e_{ij}$ to update the local geometric features $x_i^{'}$ of point $i$.

$$\alpha_{ij} = softmax\left(LeakyReLU\left(h_\Theta\left(e_{ij}\right)\right)\right) = \frac{exp\left(LeakyReLU\left(h_\Theta\left(e_{ij}\right)\right)\right)}{\sum_{j\in N_i} exp\left(LeakyReLU\left(h_\Theta\left(e_{ij}\right)\right)\right)} \tag{2}$$

$$x_i^{'} = \sum_{j\in N_i} \alpha_{ij} e_{ij} \tag{3}$$

Because different AtEdgeConv layers capture different local geometric structures, we stack two AtEdgeConv layers into the born network of DGCNN to expand the receptive field shown in Fig. 2. For each roadside objects, we take the 3D coordinates ($x$, $y$, $z$) of roadside objects as input of the point cloud branch. To align the roadside objects, we embed a spatial transform layer at the front of the network. Finally, outputs of two AtEdgeConv layers are concreted and combined into the discriminative pointwise geometric features $\mathscr{F}_p$ of the roadside objects by a max pooling layer.

$$\mathscr{F}_p = maxpool\left(\left[x_i^{\text{AtEdgeConv1}} || x_i^{\text{AtEdgeConv2}}\right]\right) \tag{4}$$

### 3.2. Global viewwise feature extraction

As local geometric features mainly describe the detailed information of local patches, aggregating them into a global descriptor of roadside objects requires progressively enlarging the receptive field, which is inefficient and has huge computational costs. In contrast, global information can be more easily captured from views by well-established CNN deep learning models. Because of the generation method of multiview images, some views are similar to each other, and some are quite different. Treating viewwise features equally across views may reduce the contribution of useful features but increase the effect of insignificant features, leading to fewer discriminative descriptors of the roadside objects. Hence, we aim to group similar views and exploit the significant group context information to reduce redundant information from multiple views. Inspired by the work of Feng et al. (2018), we stack a group-view convolutional neural network (GVCNN) to extract the discriminative global shape descriptor, including the view-group-object level features of the roadside objects, as shown in Fig. 3.

(1) View-level feature extraction

This block aims to extract two kinds of view features, $f_{v_i}$ and $f_{v_i}^{'}$, which describe different contextual information of the views for obtaining grouping scheme and group-view features. For each roadside object, we generate a set of views $\mathscr{V} = \{v_1, v_2, \ldots v_n\}$ of size 224 × 224 at equal intervals, where the grid value is marked as "1″ if it contains points, otherwise it is marked as "0″. To reduce the training time, we implement a pretrained VGG16 $C_1$-$C_4$ and $C_5$ network on each view to

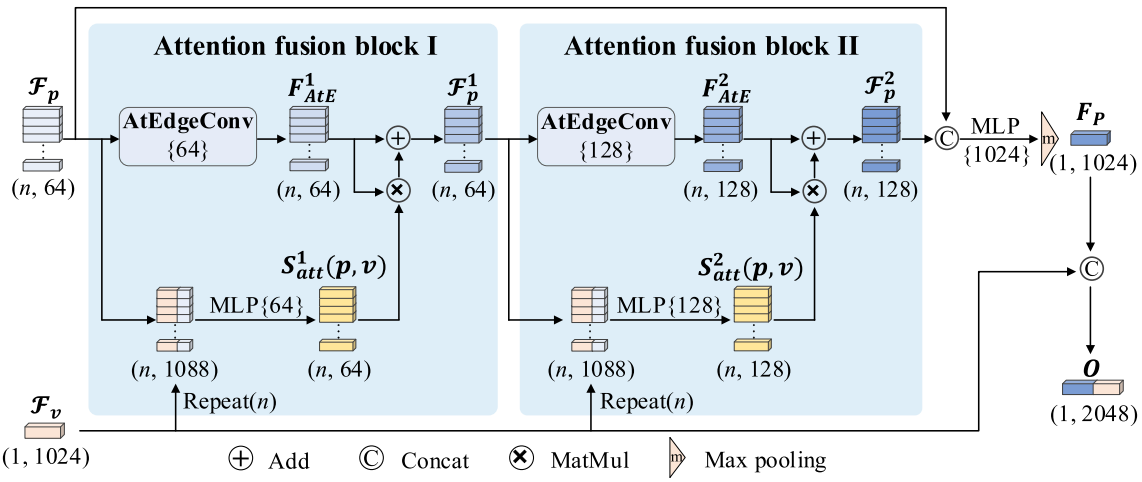



**Fig. 4.** Point-view feature attention embedding fusion network. It repeats the global viewwise features $\mathscr{F}_v$ $n$ time and concatenate it with local geometric features $\mathscr{F}_p$ to generate the attention mask, which is the guidance to build the nonlocal relations between the output feature of AtEdgeConv.

capture view features $f_{v_i}$ and $f'_{v_i}$, respectively, because of its good feature representation capabilities and reasonable network architecture (Wang et al., 2015).

(2) Group-level feature extraction

To group similar views, the first step is to measure the similarity of the views. Compared with the view features $f'_{v_i}$, the initial view features $f_{v_i}$ contain more geometric and discriminable intraclass information, which is more conducive to grouping similar views. Hence, we quantize view features $f_{v_i}$ by coefficients $\xi(v_i)$ to measure the context-based discrimination for each view:

$$\xi(v_i) = sigmoid\left(log\left(abs\left(f_{v_i}\right)\right)\right) \tag{5}$$

where $sigmoid(\bullet)$ and $log(\bullet)$ are the activation functions of the shared MLP layer and the $log$ function, respectively. $abs(\bullet)$ is a designed function that avoids the output of $sigmoid(\bullet)$ being close to 1 or 0.

To obtain the grouping scheme, we then analyse the view coefficient frequency using histogram statistics and subdivide the views into $m$ groups $\{g_1, g_2, \ldots, g_m\}$ with an interval value $\varepsilon$. A large number of views will introduce more redundant group-level features in a group. To lessen the importance of groups with a large number of views, we further calculate the group weight $w_j$, $j \in m$, according to the view coefficients $\xi(v_i)$ and the number of views $n_j$ in group $g_j$:

$$w_j = \frac{\sum Ceil\left(\xi(v_i) \bullet n_j\right)}{n_j}, v_i \in g_j \tag{6}$$

Meanwhile, the views within the same group share similar content and close discriminations. To highlight the significant information between intragroup views, an average pooling scheme is implemented on the intragroup views to extract the group-level feature $\mathscr{F}_{g_j}$ of each group.

$$\mathscr{F}_{g_j} = Ave\left(f'_{v_i}\right),\ v_i \in g_j \tag{7}$$

where $Ave(\bullet)$ denotes the function of the average pooling layer.

(3) Object-level feature extraction

Finally, we weight all group-level features $\mathscr{F}_{g_j}$ to generate object-level features $\mathscr{F}_v$ by an MLP layer. In this manner, we jointly consider the content and the discriminativity of the views and generate a discriminative viewwise descriptor for roadside objects.

$$\mathscr{F}_g = \frac{\sum_{j=1}^{m} w_j \cdot \mathscr{F}_{g_j}}{\sum_{j=1}^{m} w_j} \tag{8}$$

$$\mathscr{F}_v = mlp\left(\mathscr{F}_g\right) \tag{9}$$

where $mlp(\bullet)$ denotes the function of the MLP layer.

### 3.3. Point-view feature attention embedding fusion

To fuse different level features from two models, we introduce two attention fusion blocks shown in Fig. 4. In each attention fusion block, the global viewwise features from multiview data are used as guidance to build the nonlocal relations between the different local geometric features from the point cloud data.

Given the $l$ th layer of the attention embedding fusion block, we repeat the view features $n$ times and concatenate it with point features to adaptively generate the soft attention masks $S^l_{att}(p, v)$, which are implemented by an MLP layer and a normalization function $\zeta(\bullet)$.

$$\phi^l(p, v) = \left[\text{repeat}(\mathscr{F}_v, n) || \mathscr{F}^l_p\right] \tag{10}$$

$$S^l_{att}(p, v) = \zeta\left(mlp\left(\phi^l(p, v)\right)\right) \tag{11}$$

where $\text{repeat}(\bullet)$ denotes the features repeating operator. $\phi^l(\bullet)$ is the feature concatenating function and $mlp(\bullet)$ denotes the MLP layer.

As $S^l_{att}(p, v)$ describes the significance of different point patches, we apply it to the output of AtEdgeConv layers $F^l_{\text{AtE}}$ in a residual way to explore the more discriminative point features and restrain the useless features:

$$\mathscr{F}^l_p = F^l_{\text{AtE}} * \left(1 + S^l_{att}(p, v)\right) \tag{12}$$

where $F^l_{\text{AtE}} * S^l_{att}(p, v)$ indicates applying $S^l_{att}(p, v)$ as the soft attention mask to $F^l_{\text{AtE}}$ by element-wise multiplication.

Generally, the deeper the network means more abstract features and a larger receptive field. Therefore, we stack two attention fusion blocks to sufficiently describe the relative relationships of the different local structures of the roadside objects. In addition, we fuse the local geometric features $\mathscr{F}_p$ and the final attention-aware features $\mathscr{F}^2_p$ into a high-level global point feature $F_P$ by an MLP layer and a max pooling layer. Finally, the embedded view features $\mathscr{F}_v$ are concatenated on the last fully connected layer of the point cloud to assist the final fusion.

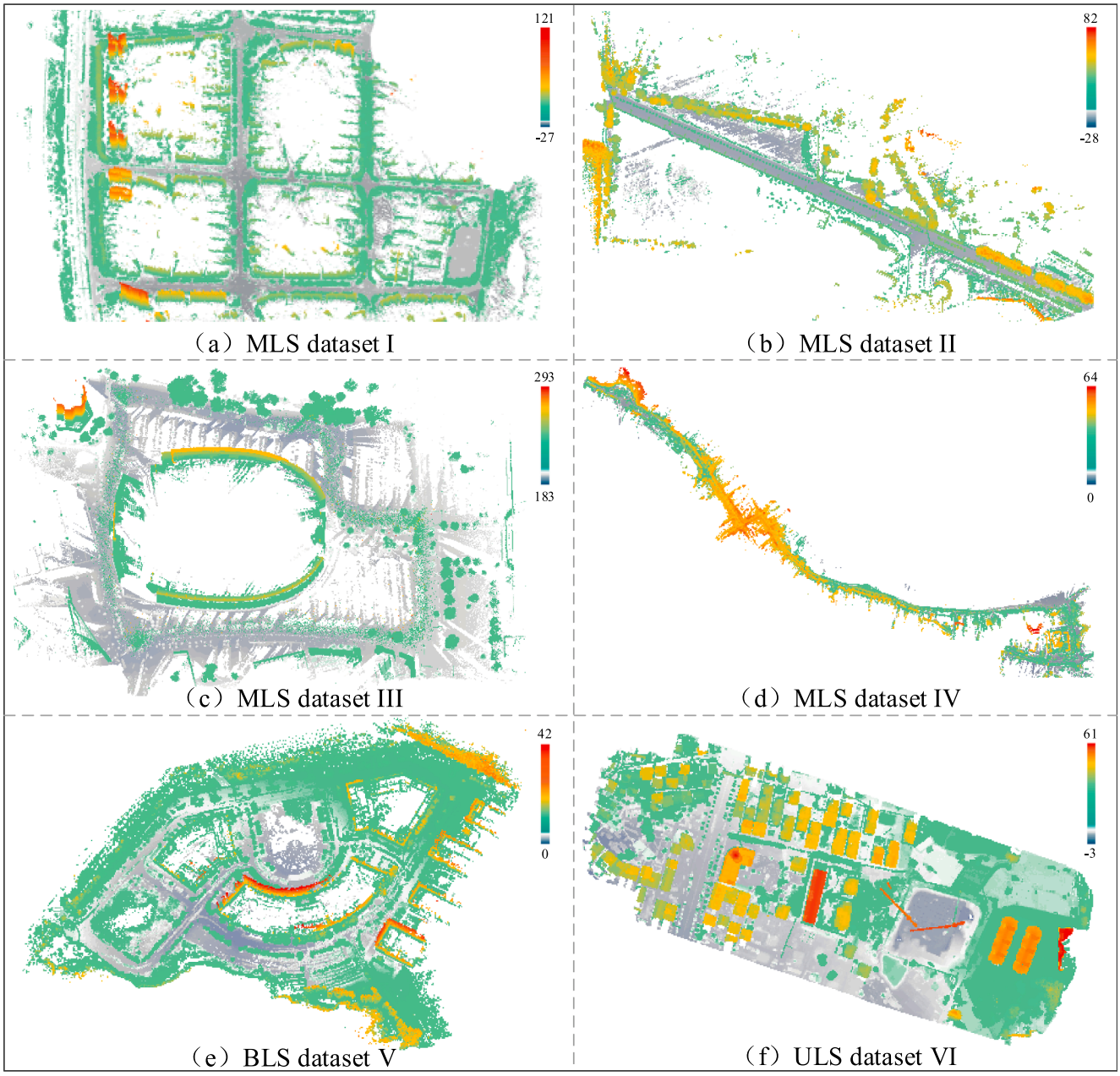



**Fig. 5.** Overview of six test datasets (point clouds colored by height data). (a) MLS dataset I; (b) MLS dataset II; (c) MLS dataset III; (d) MLS dataset IV; (e) BLS dataset V and (f) ULS dataset VI.

$$F_P = maxpool\left(mlp\left(\left[\mathscr{F}_p || \mathscr{F}_p^2\right]\right)\right) \tag{13}$$

$$O = [F_P || \mathscr{F}_v] \tag{14}$$

### 3.4. Loss function

In different road scenes, category imbalances widely exist, which has a negative impact on the performance of roadside object classification. To overcome this issue, we implement a median frequency balancing strategy and assign different category weights $w_c$ to define the loss function $\mathscr{L}_{loss}$:

$$\mathscr{L}_{loss} = \sum_{c=1}^{C} w_c y_c \ln \hat{y}_c + \lambda \|\omega\|_2^2 \tag{15}$$

$$w_c = \frac{Median\left(\left\{\frac{N_c}{\sum_{c=1}^{C} N_c} | c \in C\right\}\right)}{\frac{N_c}{\sum_{c=1}^{C} N_c}} \tag{16}$$

where $C$ represents the number of categories and $w_c$ is the weight of category $c$. $\hat{y}_c$ denotes the probability of roadside object being to category $c$, and $y_c$ is the corresponding label. $\lambda\|\omega\|_2^2$ denotes the regularization term to avoid overfitting during training. $\lambda$ and $\|\bullet\|_2$ are the regularization coefficient and $L_2$ paradigm, respectively. $\omega$ is a trainable weight vector. $N_c$ is the number of categories $c$ in the total sample $N$.

**Table 1**
Descriptions of six test datasets.

| Information | MLS dataset I | MLS dataset II | MLS dataset III | MLS dataset IV | BLS dataset V | ULS dataset VI |
|---|---|---|---|---|---|---|
| Location | Fuzhou China | Beijing China | Columbus USA | Putian China | Fuzhou China | Fuzhou China |
| Scene | Urban | Town | Urban | Urban | Urban | Town |
| Scanning systems | RIEGL VMX-450 | Trimble Mx8 | Optech Lynx | HiScan-Z | LiBackpack | DJI L1 |
| Type | MLS | MLS | MLS | MLS | BLS | ULS |
| Road length (km) (km) | 4.59 | 0.70 | 1.50 | 5.00 | 1.70 | 0.24 |
| Density (pts/m$^2$) | 170 | 90 | 124 | 79 | 123 | 728 |
| Total Points (million) | 388.42 | 20.50 | 26.01 | 76.00 | 112.75 | 108.05 |
| Off-ground objects Points (million) | 122.51 | 1.73 | 1.03 | 11.80 | 7.99 | 10.42 |
| Car | 1076 | 6 | 391 | 167 | 259 | 86 |
| Tree | 2283 | 339 | 285 | 2078 | 677 | 569 |
| Traffic pole | 580 | 56 | 161 | 578 | 103 | 23 |
| Small object | 925 | 8 | 47 | 142 | 7 | – |
| Other | 125 | – | 15 | 140 | 31 | 2 |

## 4. Experiments and analyses

### 4.1. Dataset description

#### 4.1.1. Test datasets

To evaluate the efficiency of our PGVNet, six test datasets (shown in Fig. 5) are selected to conduct experiments. These test datasets contain typical urban scenes from different regions and contain many cars, trees, traffic poles, pedestrians, bicycles and e-bicycles. The overview description of test datasets is listed in Table 1. Four MLS datasets with various characteristics, which are captured by different sensors and located in various road types with discrepant densities and integrity of point clouds. In detail, test dataset I covers the largest scene among the test datasets, and includes a considerable number of cars, trees, pedestrians, bicycles and e-bicycles. In particular, due to acquiring the point cloud on a rainy day, many pedestrians were wearing umbrellas and had raincoats or other protection from rain. In test dataset III, point clouds are very sparse, and the completeness of objects is not very good. A large number of trees and cars are only partially preserved, which makes object identification more difficult. Test dataset IV contains many trees and traffic poles with different shapes. Due to occlusion, many roadside objects are incomplete. To further evaluate the proposed PGVNet model's generalizability for multiple data acquisition methods, test dataset V acquired by the BLS system and test dataset VI acquired by the ULS system are chosen. Test dataset V and VI contain numerous trees and cars but there is a large amount of noise around them due to the unsatisfactory performance of the device. In particular, most objects in test dataset VI lose their lower part of the point clouds and become severely incomplete. The diversity of roadside object types, as well as data quality issues such as object incompleteness and the noise in these datasets, present a huge challenge to our PGVNet and are suitable to verify its effectiveness. We manually labelled the cars, trees, traffic poles, pedestrians, bicycles, e-bicycles and other objects in six test datasets as the ground truth (see Table 1). As pedestrians, bicycles and e-bicycles are important small roadside objects that appear together frequently, they are regarded as belonging to the small object category and given the same label.

#### 4.1.2. Training datasets and samples

Due to the lack of available training samples, we select four datasets (shown in Fig. 6) to construct the training sample set. These four datasets were obtained by the MLS system, and Table 2 shows their brief descriptions. The training dataset I is from the public dataset Paris-Lille-3D (Roynard et al., 2018b) and its point density is very high. Other training datasets are common MLS point clouds with considerable roadside objects, which contribute to the completeness of the training samples. We use the segmentation method by Fang et al. (2020) to extract the individual roadside objects from these training datasets, and manually labeled the training samples illustrated in Fig. 6. The number of each category is shown in Table 2. For each class, the number of training samples are expanded to 550 including 500 training samples and 50 validation samples, through data enhancement operations such as panning, rotating and adding dithering noise.

### 4.2. Implementation and parameters

The proposed model is implemented on the TensorFlow1.8 platform with a NVIDIA GeForce GTX 1080ti GPU (11 GB RAM total). During training, the stochastic gradient descent optimization strategy with momentum is utilized to update the model. The initial learning rate, momentum and decay rate are set at 0.001, 0.9 and 0.7, respectively. An entire training process consists of 30 epochs with a batch size of 16. In addition, we use a model fine-tuning-based method to reduce the sample and hardware requirements, which initialize the parameters of global viewwise feature extraction network by a pretrained VGG16 model and then fine-tuned them with training samples. In this way, the global viewwise feature extraction network is stronger at the early training stage but the local geometric feature extraction network is weaker. Thus, the parameters of the global viewwise feature extraction network are frozen for ten epochs at an early stage and then updated together with entire network's parameters.

As a multimodal fusion approach, our PGVNet uses two modalities of data: 2048 points with 3D coordinates and 8 views as input. For each roadside object, its point cloud representation is obtained by the farthest point sampling (FPS) operation, while the multi-views representation is generated by projecting them horizontally at equal angles. Precision (P), recall (R), quality (Q), F1-score (F1) are calculated in each category and then overall accuracy (OA) are employed over all samples to evaluate the results in this paper, and they are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{17}$$

$$R = \frac{TP}{TP + FN} \tag{18}$$

$$Q = \frac{TP}{TP + FP + FN} \tag{19}$$

$$F1 = \frac{2PR}{P + R} \tag{20}$$

$$OA = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} N_i} \tag{21}$$

where TP denotes the number of true positives, FP denotes the number

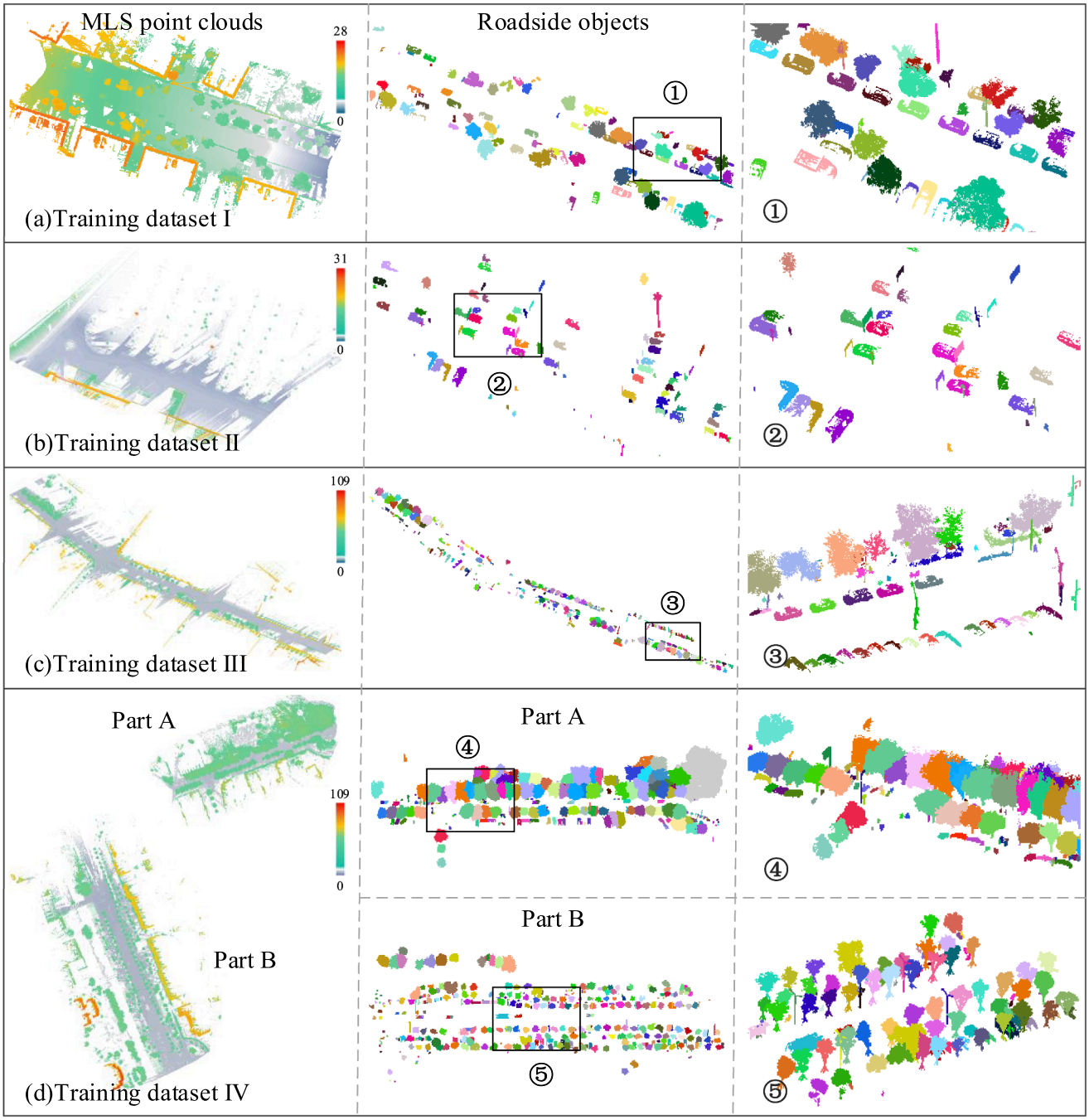



**Fig. 6.** Four datasets used to generate training samples. The first column lists the four training datasets (colored by height data). The second column is the corresponding individual roadside objects for making samples, which are indicated by different colors. The third column is the close-up views of the selected regions.

of false positives, FN denotes the number of false negatives, $C$ represents the number of categories, $TP_i$ denotes the number of true positives in the $i$ th category, and $N_i$ is the number of samples in the $i$ th category.

In the PGVNet, the parameter $k$ at AtEdgeConv, and interval value $\varepsilon$ in the view-grouping scheme intervals are important to the model performance, where $k$ influences the local feature extractions of the object and $\varepsilon$ influences the global view feature extractions. To acquire the optimal parameters, we use the OA as the evaluation metric to conduct comparison experiments on the training datasets. First, we experiment with different values of $k$ under an interval value $\varepsilon$ of 0.1, and the results are shown in Table 3. When the parameter $k$ was set to 10, the model achieved the best result, so we selected 10 as the optimal value of $k$. Then, freezing parameter $k$, we conducted some experiments with different values of $\varepsilon$. We observed that our PGVNet obtained the best performance when the parameter $\varepsilon$ was set to 0.1. Therefore, we set the parameters $k$ and $\varepsilon$ to 10 and 0.1, respectively, in all the experiments.

### 4.3. Result of roadside object recognition

After training PGVNet with the optimal parameter configuration, we evaluated it on the selected test datasets. According to the segmentation method by Fang et al. (2020), all roadside objects are segmented and generated into two modalities as the input of the trained PGVNet for category prediction. To clearly illustrate the validated results of each test dataset, we list some subplots of the results from a detailed view (see Fig. 7). We can see that despite the trees, cars, traffic poles and small

**Table 2**
Brief information of training datasets.

| Information | Training dataset I | Training dataset II | Training dataset III | Training dataset IV |
|---|---|---|---|---|
| Location | Ajaccio France | Corsico Italy | Toronto Canada | Fuzhou China |
| Scene | Urban | Urban | Urban | Urban |
| Scanning systems | L3D2 | Optech Lynx | Optech Lynx | RIEGL VMX-450 |
| Type | MLS | MLS | MLS | MLS |
| Road length (km) | 0.20 | 0.13 | 0.40 | 0.49 |
| Density (pts/$m^2$) | 839 | 138 | 196 | 170 |
| Total points (million) | 10.00 | 8.55 | 16.19 | 41.28 |
| Off-ground objects points (million) | 1.58 | 0.39 | 0.68 | 12.46 |
| Car | 4 | 45 | 71 | 89 |
| Tree | 50 | 38 | 95 | 337 |
| Traffic pole | 43 | 14 | 89 | 60 |
| Small object | 11 | 3 | 30 | 227 |
| Other | 23 | 8 | 59 | 13 |

objects in dataset I with large variances in shape, most of them were correctly recognized. For incomplete and sparse roadside objects such as trees, traffic poles and cars in datasets II, III and IV, the proposed method also performs stably, and most of the objects are classified into the correct categories. In addition, the test objects are plagued by noise, especially in dataset V but the result shows that our method can successfully overcome the interference. With dataset VI, the ULS scene, our method still works well even if the objects only have part of the geometric structure. The result shows that the proposed method shows an exciting performance for the test datasets. In addition, the proposed method is able to overcome data quality impacts and effectively identify most of the roadside objects in real-world scenarios.

However, it should be pointed out that there do exist some misclassified cases. Fig. 8(a) shows that the tree trunks without crowns are misclassified as traffic poles due to significant geometric similarities. In the scene of Fig. 8(b) and Fig. 8(c), the low trees are detected as traffic poles, small objects, or other objects, respectively. In Fig. 8(d), the incomplete shrubs are recognized as cars and other objects. For this situation, two possible reasons are considered. First, due to the lack of corresponding samples, some test objects are weakly defined and classified into error categories. Additionally, the occlusion and sparseness of the point cloud could cause a problem that some objects with similar geometric shapes may share some key discriminating features, which causes those objects to be classified into the faulty class.

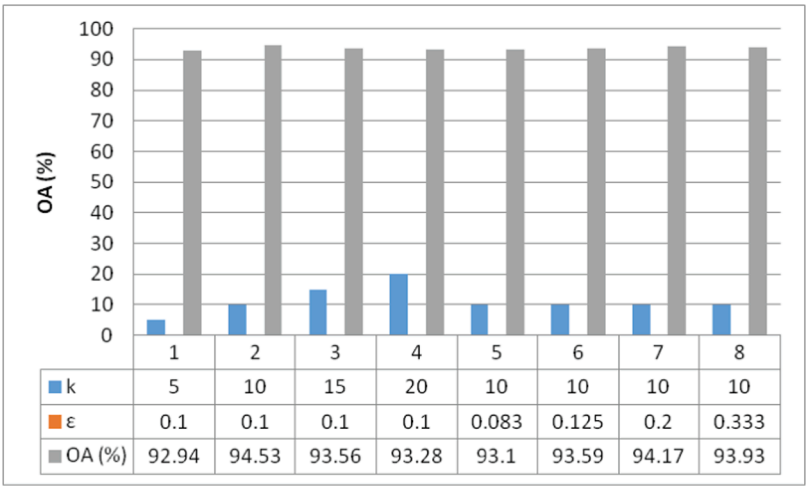
To quantitatively assess the performance of our method, the precision (P), recall (R), quality (Q) and F1-score (F1) of each category were calculated and are listed in Table 4. In particular, the method performs better on the car, tree, and traffic pole classes and achieves an overall F1-score of 96.49 %, 97.70 % and 95.50 %, respectively. Because most cars, trees and traffic poles have various shapes, our method can extract distinguishable deep features with a few training samples. Thus, the results of three categories, including car, tree and traffic pole, in all test datasets are satisfactory, especially benefitting from the relatively complete shapes and reliable features. The F1-score of the car and tree categories in test datasets III and V is over 99 %. As the slender trees were easily misclassified as traffic poles, the precision of the traffic pole in dataset IV is only 89.97 %. Because the training samples not contain special tree samples without crown, some incomplete trees were misclassified as cars in dataset VI, which led to a low precision value of the car (70 %) and recall of the tree (89.46 %).

In comparison, the small object class achieved a slightly worse performance, and the overall F1-score was 89.45 %. Due to objects in the small object classes that vary widely in shape, they are sparser and easily obscured due to their small size. Thus, their deeper features are obscure and more likely to be confused with other objects such as traffic poles, low trees and incomplete cars. This also leads to the low precision of the small objects in test datasets II, III, IV and V. In addition, the pedestrians being classified as traffic poles and other objects also caused the lower overall F1-score of the small objects. Remarkably, despite not retraining the model, good classification results were still achieved in the ULS scene, with F1-score values of 81.55 %, 94.35 % and 97.78 % for cars, trees and traffic poles, respectively. This result shows the applicability and generalizability of our method for different source data.

### 4.4. Accuracy performance comparison

To demonstrate the advantages of the proposed method, we compare the proposed method with eight existing methods that work well in recognizing 3D objects. The methods include three multiview methods, including DBN (Fang et al., 2020), MVCNN (Su et al., 2015) and GVCNN (Feng et al., 2018); four mainstream point-based methods, including PointNet (Qi et al., 2017a), DGCNN (Wang et al., 2019b), KPConv (Thomas et al., 2019) and MinkowskiNet (MinkNet) (Choy et al., 2019); and one multimodal method, PVNet (You et al., 2018). For all the comparative models, the public codes are implemented in Python with the same computing resources and datasets. In the experiments, we use

**Table 3**
Classification performance (OA) on the training datasets of our PGVNet for different parameter configurations.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| k | 5 | 10 | 15 | 20 | 10 | 10 | 10 | 10 |
| ε | 0.1 | 0.1 | 0.1 | 0.1 | 0.083 | 0.125 | 0.2 | 0.333 |
| OA (%) | 92.94 | 94.53 | 93.56 | 93.28 | 93.1 | 93.59 | 94.17 | 93.93 |

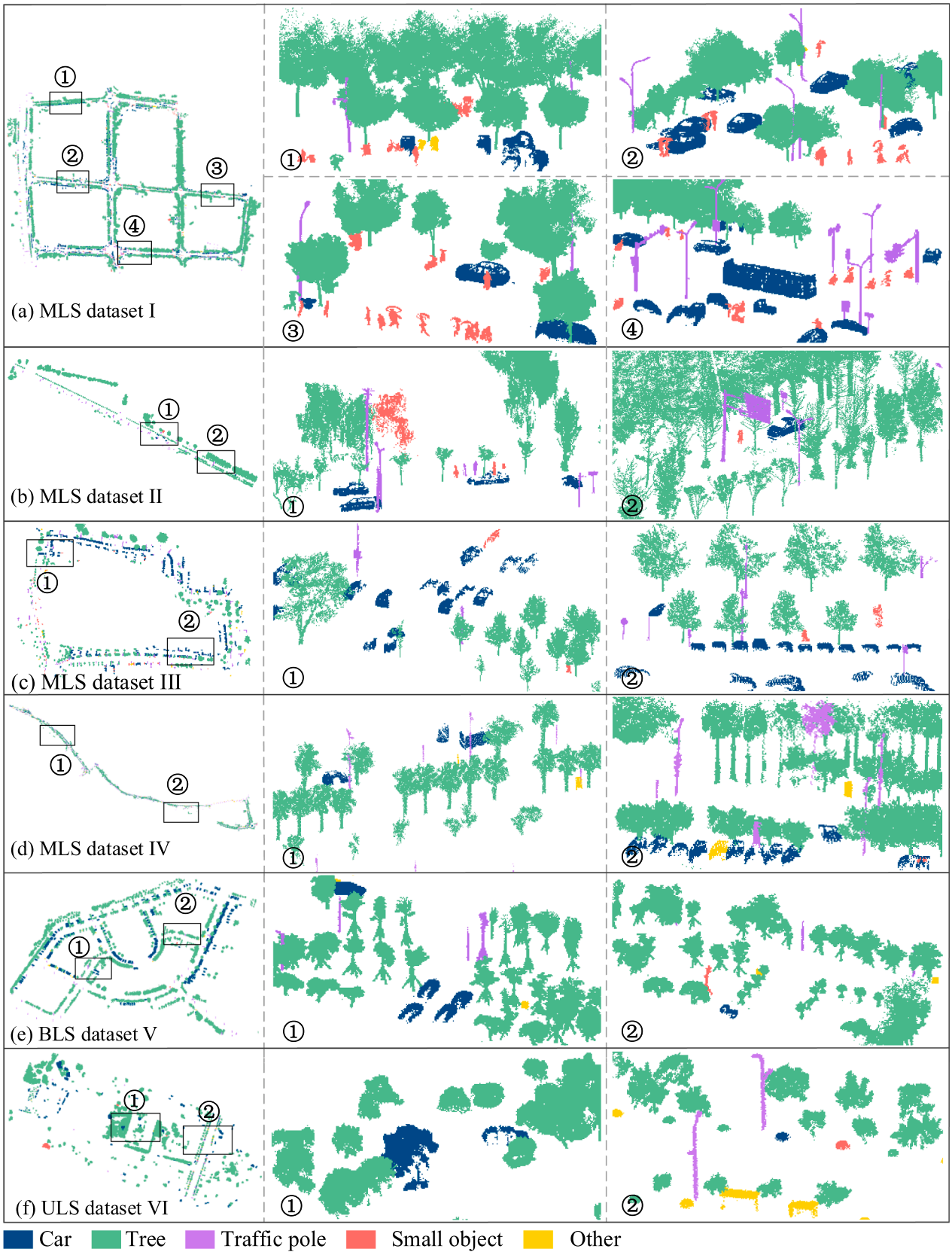



**Fig. 7.** Classification performance of our PGVNet on six test datasets. The first column lists the roadside object recognition results of six test datasets. The second and third columns present some selected regions of the results in a close-up view.

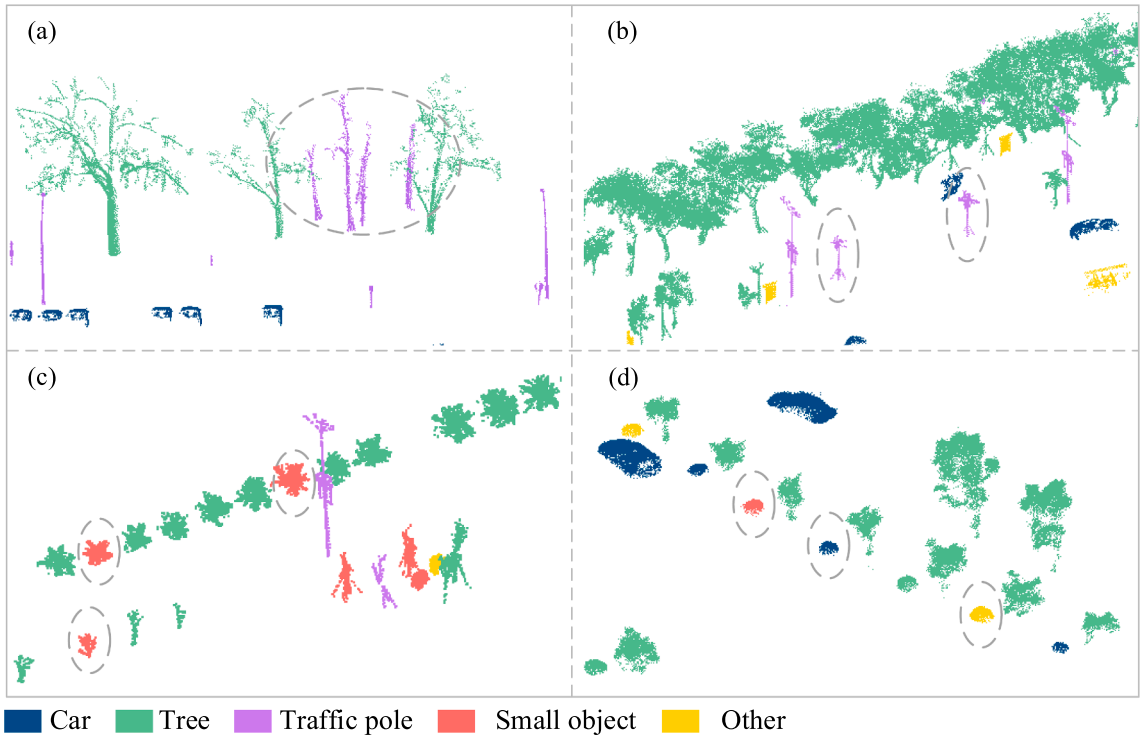



**Fig. 8.** Typical misclassified scenes on the test datasets. (a) four thin trees are misclassified into traffic pole class; (b) two trees without crown data are misclassified into traffic pole class; (c) three low trees are misclassified into small object class and (d) three shrubs are misclassified.

**Table 4**
Accuracy of roadside object classification results.

| Dataset | Class | TP | FN | FP | P (%) | R (%) | Q (%) | F1 (%) |
|---|---|---|---|---|---|---|---|---|
| MLS | Car | 1,044 | 32 | 25 | 97.66 | 97.03 | 94.82 | 97.34 |
| dataset I | Tree | 2,261 | 23 | 50 | 97.84 | 98.99 | 96.87 | 98.41 |
| | Traffic pole | 575 | 5 | 20 | 96.64 | 99.14 | 95.83 | 97.87 |
| | Small object | 836 | 88 | 32 | 96.31 | 90.48 | 87.45 | 93.30 |
| MLS | Car | 372 | 19 | 3 | 99.20 | 95.14 | 94.42 | 97.13 |
| dataset II | Tree | 273 | 12 | 0 | 100.00 | 95.79 | 95.79 | 97.85 |
| | Traffic pole | 154 | 7 | 12 | 92.77 | 95.65 | 89.02 | 94.19 |
| | Small object | 44 | 3 | 16 | 73.33 | 93.62 | 69.84 | 82.24 |
| MLS | Car | 6 | 0 | 0 | 100.00 | 100.00 | 100.00 | 100.00 |
| dataset III | Tree | 338 | 1 | 0 | 100.00 | 99.71 | 99.71 | 99.85 |
| | Traffic pole | 56 | 0 | 3 | 94.92 | 100 | 94.92 | 97.39 |
| | Small object | 5 | 3 | 1 | 83.33 | 62.50 | 55.56 | 71.43 |
| MLS | Car | 159 | 8 | 13 | 92.44 | 95.21 | 88.33 | 93.81 |
| dataset IV | Tree | 1,987 | 91 | 40 | 98.03 | 95.62 | 93.81 | 96.81 |
| | Traffic pole | 556 | 22 | 62 | 89.97 | 96.19 | 86.88 | 92.98 |
| | Small object | 106 | 35 | 45 | 70.20 | 75.18 | 56.99 | 72.60 |
| BLS | Car | 257 | 2 | 0 | 100.00 | 99.23 | 99.23 | 99.61 |
| dataset V | Tree | 674 | 3 | 3 | 99.56 | 99.56 | 99.12 | 99.56 |
| | Pole | 101 | 2 | 4 | 96.19 | 98.06 | 94.39 | 97.12 |
| | Small object | 5 | 2 | 1 | 83.33 | 71.43 | 62.50 | 76.92 |
| ULS | Car | 84 | 2 | 36 | 70.00 | 97.67 | 68.85 | 81.55 |
| dataset VI | Tree | 509 | 60 | 1 | 99.80 | 89.46 | 89.30 | 94.35 |
| | Traffic pole | 22 | 1 | 0 | 100.00 | 95.65 | 95.65 | 97.78 |
| | Small object | 0 | 0 | 9 | – | – | – | – |
| Overall | Car | 1,922 | 63 | 77 | 96.15 | 96.83 | 93.21 | 96.49 |
| | Tree | 6,042 | 190 | 94 | 98.47 | 96.95 | 95.51 | 97.70 |
| | Traffic pole | 1,464 | 37 | 101 | 93.55 | 97.53 | 91.39 | 95.50 |
| | Small object | 996 | 131 | 104 | 90.55 | 88.38 | 80.91 | 89.45 |

2048 points and 8 or 3 views (DBN) to respect a roadside object in the training and test stages.

Fig. 9 presents the PR curves of the different methods on the test data. It shows that our method obtains better results than the other methods, in which the precision is 90 % and the recall of our method is over 95 %. As shown in Fig. 9(d), the precision of our method is over 95 % when the recall is 90 %, which indicates that our method is stable on dataset IV. Despite the objects in dataset VI being very incomplete, our method still performs better than the comparative methods, especially the methods of DBN and PointNet. On dataset VI, our method significantly outperforms the state-of-the-art on this task. Our method achieves a precision of approximately 90 % when the recall is 90 %, while the DBN, MinkNet and PointNet are below 40 %, as shown in Fig. 9(f).

In addition, we employ quality, F1-score, and overall accuracy as

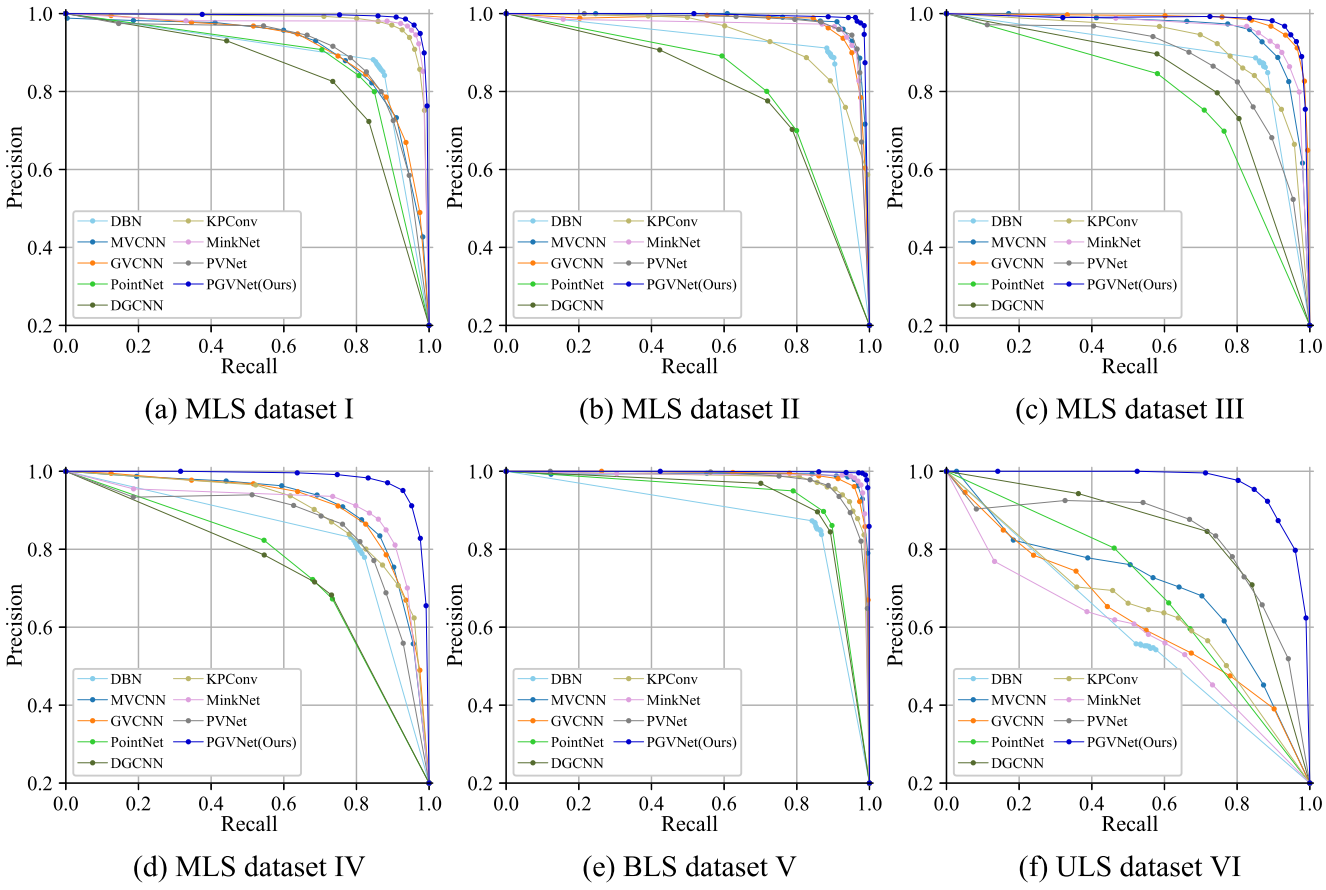



**Fig. 9.** PR curves obtained by different methods in the six test datasets.

**Table 5**
Accuracy comparison of our method with some of the existing methods.

| Method | Input | Car | | Tree | | Traffic pole | | Small object | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Q(%) | F1(%) | Q(%) | F1(%) | Q(%) | F1(%) | Q(%) | F1(%) | OA(%) |
| DBN | 3 views | 79.13 | 88.35 | 83.32 | 90.90 | 83.40 | 90.95 | 52.53 | 68.88 | 86.75 |
| MVCNN | 8 views | 81.53 | 89.83 | 83.94 | 91.27 | 85.25 | 92.04 | 52.87 | 69.17 | 87.62 |
| GVCNN | 8 views | 83.46 | 90.98 | 86.99 | 93.04 | 69.16 | 81.77 | 64.12 | 78.14 | 88.34 |
| PointNet | 2048 points | 78.02 | 87.65 | 84.98 | 91.88 | 82.82 | 90.60 | 44.14 | 61.25 | 86.41 |
| DGCNN | 2048 points | 77.07 | 87.05 | 86.28 | 92.64 | 85.25 | 92.04 | 45.62 | 62.66 | 86.93 |
| KPConv | 2048 points | 83.39 | 90.94 | 83.24 | 90.85 | 86.84 | 92.96 | 58.14 | 73.53 | 88.44 |
| MinkNet | 2048 points | 82.40 | 90.35 | 88.61 | 93.96 | 85.73 | 92.32 | 67.01 | 80.24 | 90.50 |
| PVNet | 2048 points + 8 views | 83.95 | 91.28 | 88.68 | 94.00 | 90.00 | 94.74 | 57.22 | 72.79 | 90.27 |
| PGVNet (Ours) | 2048 points + 8 views | **93.21** | **96.49** | **95.51** | **97.70** | **91.39** | **95.50** | **80.91** | **89.45** | **95.76** |

metrics to assess the accuracy of the comparative models. Table 5 presents the performance of each method on four roadside object categories in all test datasets. To further illustrate the classification performance of the different models, we show the comparative results in Figs. 10–12 for each method tested in three challenging typical scenarios from test dataset I, V and VI, respectively. Overall, the quantitative and illustrative results in Table 5 and Figs. 10–12 indicate that our method achieves a remarkable classification performance, with an OA of 95.76 %. The promising performance of our proposed PGVNet can be attributed to the following reasons: first, the proposed method jointly exploits the advantages of the point cloud and the multiview data for 3D roadside object representation. Second, the proposed grouping views model enables us to explore the relationship among views and capture the global object-level descriptor of the roadside objects. Third, our attention fusion module can work as a feature selector and quantify global features from the multiview data into a soft attention mask to highlight important features and suppress the useless features such as noise from the point clouds, which generates more powerful and discriminative features for roadside object recognition.

(1) Comparative studies with view-based methods

Compared with view-based methods, our PGVNet obtains improvements of OA of 9.0 %, 8.1 % and 7.4 % over DBN, MVCNN and GVCNN, respectively. Benefiting from exploiting high-order global features, three multiview-based methods achieve better performance on intact large roadside objects than on small fragmented ones. As shown in Figs. 10-12, multiview-based methods are able to classify most cars, trees and traffic poles with complete shapes. In the DBN, roadside objects are translated into three binary images without some crucial

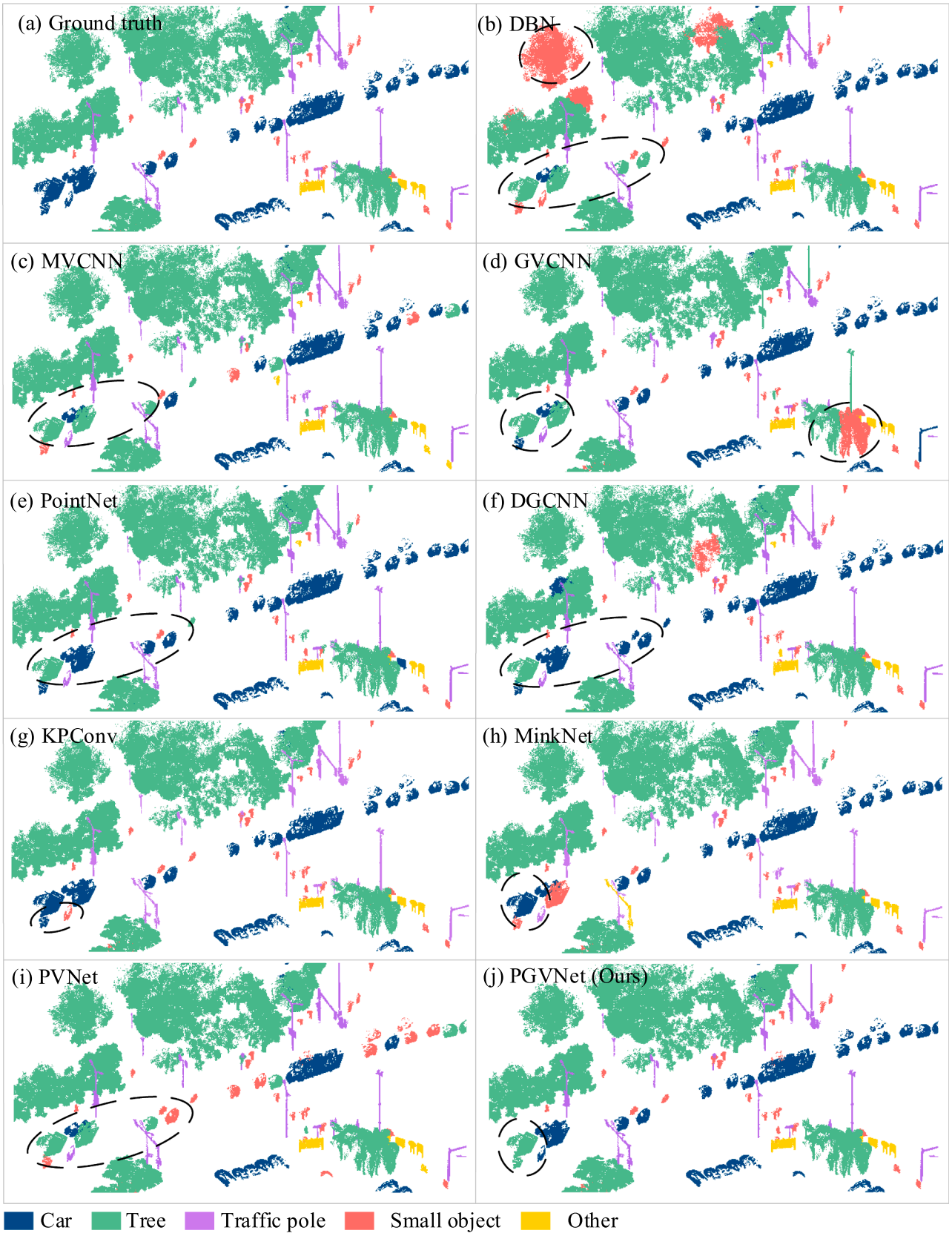



**Fig. 10.** Comparison of our method with some existing methods on the selected scene of MLS dataset I. According to the ground truth, the misclassified cars and trees in each subgraph are circled out for comparison.

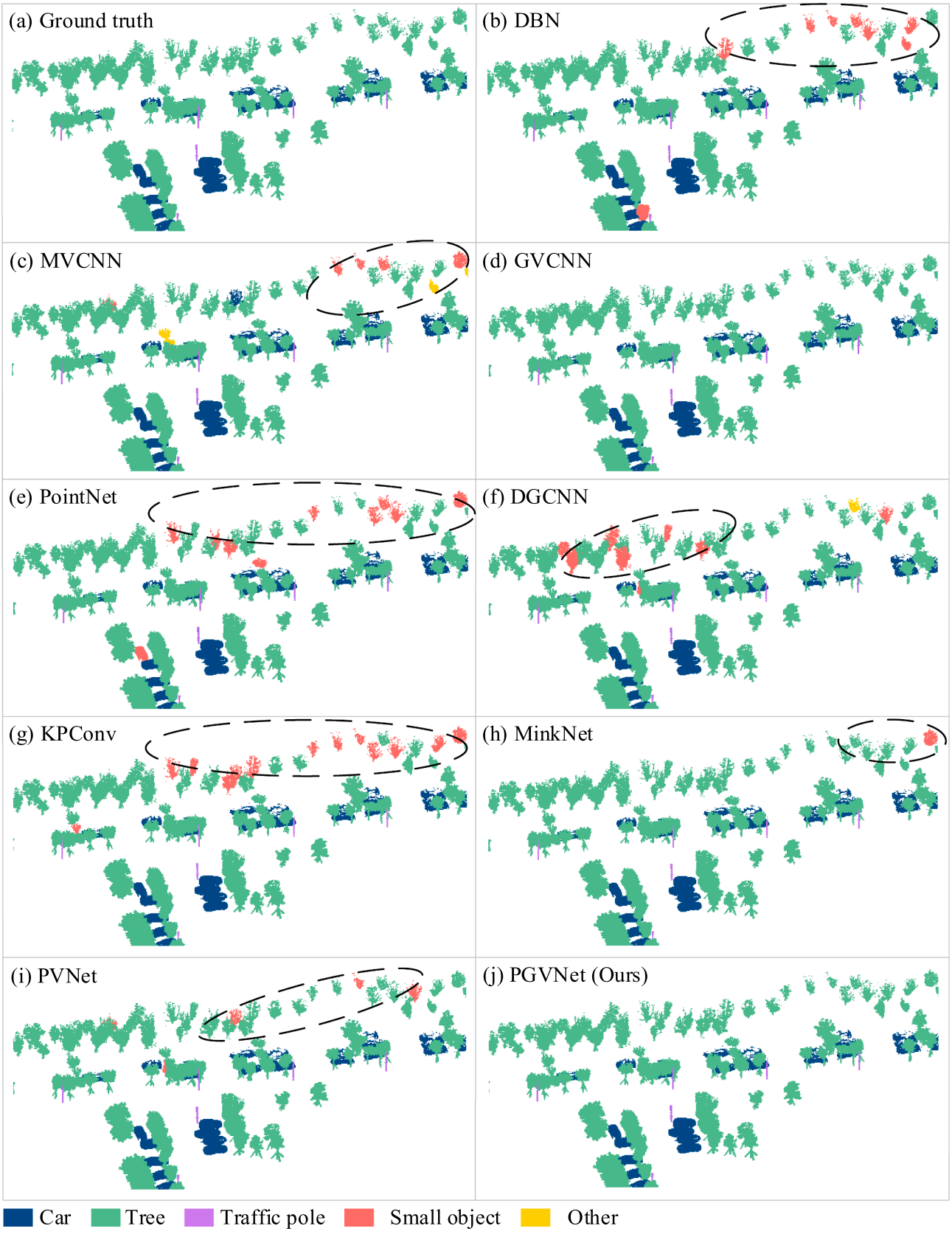



**Fig. 11.** Comparison of our method with some existing methods on the selected scene of BLS dataset V. Some trees are falsely classified into the small object class by the comparative methods and circled out in subgraphs (b), (c), (e)-(i).

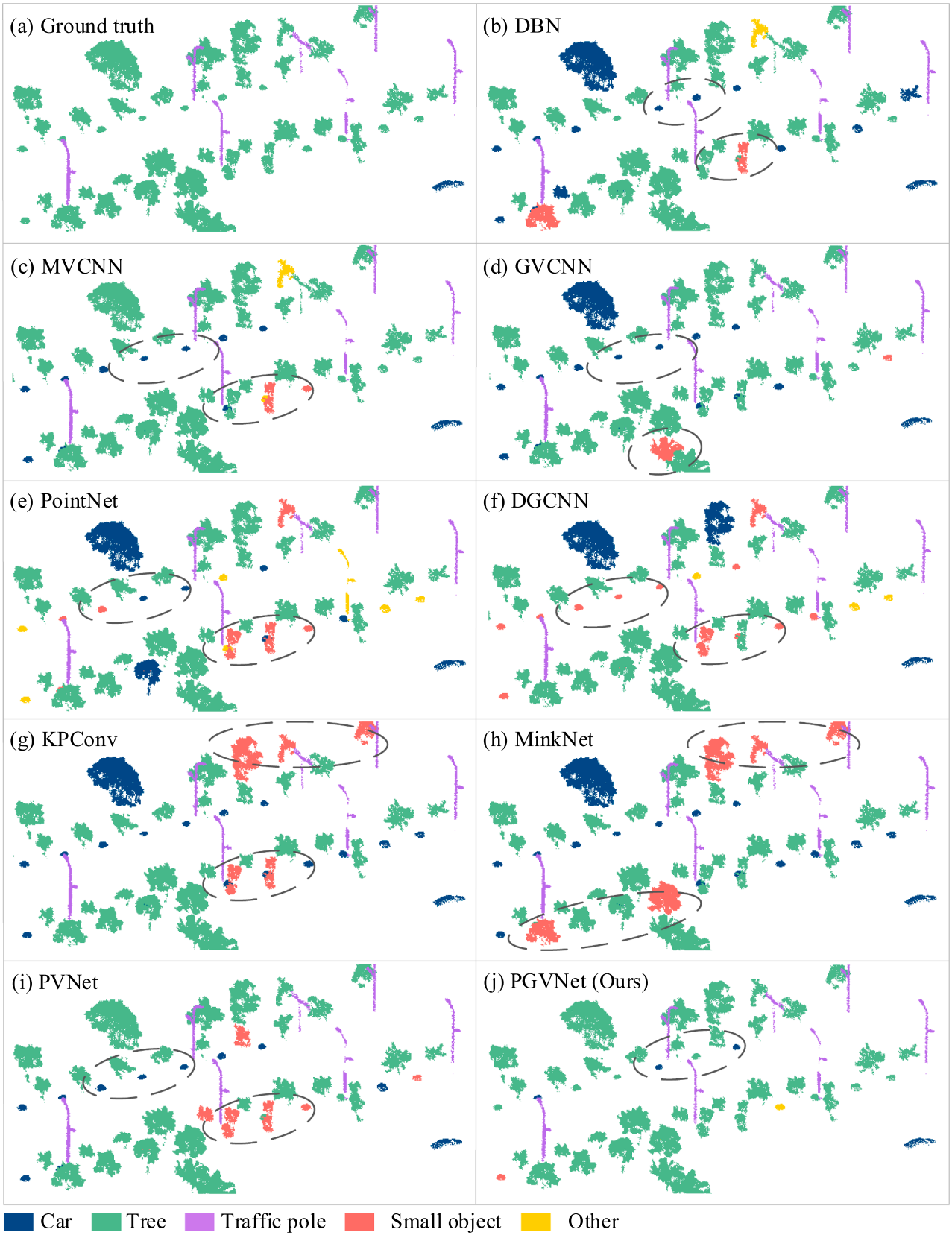



**Fig. 12.** Comparison of our method with some existing methods on the selected sense of ULS dataset VI. Some trees are falsely classified into the small object or car class and circled out in subgraphs.

**Table 6**
Time comparison of our method with some of the existing methods.

| Method | Input | Training stage(h) | | Test stage (h) | | Total |
|---|---|---|---|---|---|---|
| | | Data preprocessing | Model training | Data preprocessing | Prediction | |
| DBN | 3 views | 0.20 | 0.02 | 2.35 | 0.01 | 2.58 |
| MVCNN | 8 views | 0.51 | 2.74 | 7.03 | 0.14 | 10.42 |
| GVCNN | 8 views | 0.51 | 3.25 | 7.03 | 0.14 | 10.93 |
| PointNet | 2048 points | 0.04 | 3.99 | 0.38 | 0.04 | 4.45 |
| DGCNN | 2048 points | 0.04 | 7.96 | 0.38 | 0.06 | 8.44 |
| KPConv | 2048 points | 0.06 | 4.18 | 0.68 | 2.60 | 7.52 |
| MinkNet | 2048 points | 0.04 | 0.58 | 0.38 | 0.01 | **1.01** |
| PVNet | 2048 points + 8 views | 0.57 | 1.76 | 7.40 | 0.14 | 9.87 |
| PGVNet (Ours) | 2048 points + 8 views | 0.57 | 1.42 | 7.40 | 0.20 | 9.59 |

**Table 7**
Accuracy comparison of our proposed PGVNet model with different ablation models on the test dataset.

| Method | Input | Car | | Tree | | Traffic pole | | Small object | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Q(%) | F1(%) | Q(%) | F1(%) | Q(%) | F1(%) | Q(%) | F1(%) | OA(%) |
| $PGVNet_{point}$ | 2048 points | 80.28 | 89.06 | 88.76 | 94.05 | 86.62 | 92.83 | 50.27 | 66.91 | 88.21 |
| $PGVNet_{no\_A}$ | 2048 points + 8 views | 88.60 | 93.96 | 94.11 | 96.97 | 90.28 | 94.89 | 77.30 | 87.20 | 93.94 |
| $PGVNet_{1\_A}$ | 2048 points + 8 views | 92.15 | 95.91 | 94.70 | 97.28 | 90.67 | 95.11 | 78.82 | 88.16 | 95.13 |
| PGVNet (Ours) | 2048 points + 8 views | **93.21** | **96.49** | **95.51** | **97.70** | **91.39** | **95.50** | **80.91** | **89.45** | **95.76** |

information, so many small trees and cars (see Fig. 10 and Fig. 11) are misclassified. The MVCNN and GVCNN input more detailed view information (8 views) but fail to group some small objects. Without the help of the local geometric features, the MVCNN and GVCNN often confuse incomplete trees with pedestrians (scenes in Fig. 10 and Fig. 12). Remarkably, benefiting from the grouping views strategy, the GVCNN performed better than the MVCNN with the same view data, even on small tree classification (scenes in Fig. 11), which validates the effectiveness of the grouping scheme strategy.

(2) Comparative studies with point-based methods

Compared with classical point-based methods, PointNet and DGCNN, our PGVNet outperforms them by 9.3 % and 8.8 %, respectively. For the state-of-the-art point-based methods, KPConv and MinkNet, our PGVNet also outperforms them by 7.32 % and 5.26 % in terms of OA, respectively. Limited by the receptive field, point-based methods have difficulty constructing the nonlocal correlation between points with a large distance and cause confusion between roadside objects with similar geometric structures, such as big trees, incomplete cars and small objects (scenes in Fig. 11 and Fig. 12). Among the point-based deep learning methods, MinkNet obtains the best performance. Comparatively, MinkNet shows great superiority in the small tree (scenes in Fig. 11) and small object recognition, due to sparse tensors and the generalized sparse convolution to improve expressiveness and generalizability for high-dimensional spaces. Unexpectedly, MinkNet does not work well on ULS point clouds and gains lower PR curves on test dataset VI compared with PointNet, DGCNN and KPConv.

(3) Comparative studies with the multimodal method

As the first deep method that joins both point cloud and multiview data for 3D shape recognition, PVNet achieves a good performance among the eight comparative methods. Compared with PVNet, our PGVNet with the view-group-object grouping scheme outperforms it over 5.5 % OA on roadside objects. Especially for small object categories, including pedestrians, bicycles and e-bicycles, our PGVNet outperforms PVNet by 16.6 % in terms of F1-score. The main reason may be that for the multiview branch, PVNet used the MVCNN as the basic view model, which treats all views equally to capture the global features and ignores the difference between views, leading to redundant information and affecting the distinguishability of the final features. Our PGVNet contains a group module to group similar views and assigns different weights for pooling the viewwise features in the different groups. Thus, a certain number of pedestrians, bicycles and e-bicycles misclassified by comparative studies (see Figs. 10–12) also correctly classified by the PVNet method.

### 4.5. Time efficiency comparison

Table 6 shows the time requirements of different methods during the training and test stages. In the data preprocessing, the point subsampling process is quick while the view generation process is relatively more time-consuming. The view generation process costs about 0.5 h and 7 h on training samples and test roadside objects, respectively.

As shown in Table 6, Minknet shows a time superiority with less computational cost. It is evident that our method requires more data preprocessing time, when the training time is at an intermediate level and lies between the time cost of point-based method and multiview-based method, which indicates that our method can accelerate the learning process of point cloud features by fusion view features and can have a better time efficiency over other methods.

From the perspective of time efficiency, the DBN method with 3 views as input is incomparable. In 8 view comparisons, our PGVNet has the shortest training time, indicating that it has a competitive time efficiency. MVCNN nearly double the training time compared with our proposed method. Compared with GVCNN, the training time of our PGVNet is nearly 3 times lower. Compared with the point-based methods like PointNet and KPConv, our PGVNet takes only less than 35 % of their training time. All discussed above illustrates that our proposed method shows an advantage over the current mainstream methods on training efficiency, which means it can operate at a high-

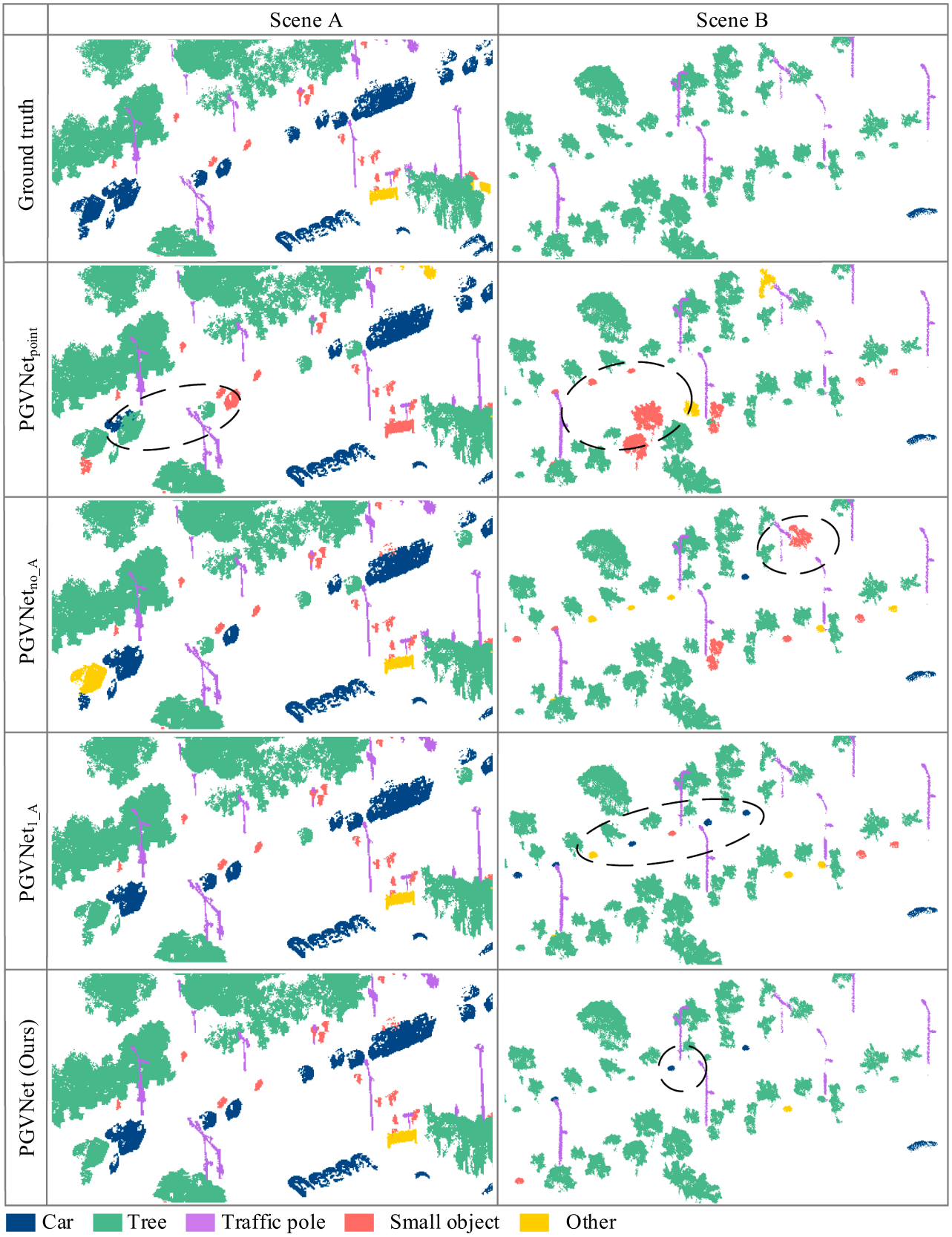



**Fig. 13.** Ablation models comparison on the selected scenes. The first column lists comparison results on Scene A. The second column lists the comparison results on Scene B. For each column, top to bottom figures are ground truth, the result of PGVNet$_{point}$, the result of PGVNet$_{no_A}$, the result of PGVNet$_{l_A}$ and our result, respectively.

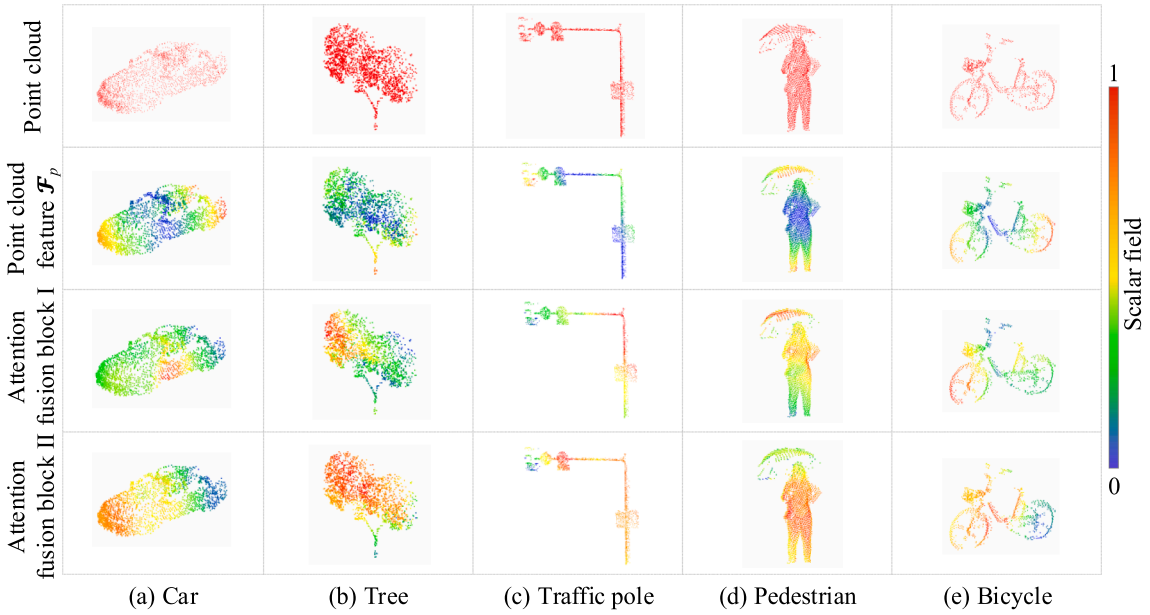



**Fig. 14.** Visualization of the point cloud local features and the attention coefficients for the attention fusion block in PGVNet. We list five examples from left to right. For each sample, from top to bottom, we illustrate the color changes from blue to red in the point cloud local features $\mathcal{F}_p$, and two attention fusion module coefficients indicate the features and weights of the different points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 8**
Our segmentation results on the Paris-Lille-3D dataset.

| Class | Ground | Building | Roadside object | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Total | Car | Tree | Pole | Pedestrian | Others |
| IoU (%) | 97.79 | 97.16 | 94.81 | 98.37 | 99.97 | 98.34 | 97.89 | 93.65 |

**Table 9**
Per class accuracy of top-5 methods and our PGVNet results on the Paris-Lille-3D dataset.

| Name | Av IoU | Ground | Building | Pole | Bollard | Trashcan | Barrier | Pedestrian | Car | Natural (tree) |
|---|---|---|---|---|---|---|---|---|---|---|
| KP-Pyramid | 83.00 | 99.50 | 97.40 | 75.50 | 81.20 | 62.80 | 72.80 | 68.70 | **96.70** | 92.80 |
| PyramidPoint | 82.90 | 99.60 | 97.10 | 74.60 | 84.30 | 56.00 | 65.90 | 79.10 | 95.10 | 93.90 |
| FKAConv | 82.70 | 99.60 | 98.10 | 77.20 | 91.10 | 64.70 | 66.50 | 58.10 | 95.60 | 93.90 |
| KP-FCNN | 82.00 | 99.50 | 94.00 | 71.30 | 83.10 | 78.70 | 47.70 | 78.20 | 94.40 | 91.40 |
| FG-Net | 81.90 | 99.50 | 94.60 | 69.30 | 83.10 | 75.00 | 47.20 | **79.80** | 96.20 | 92.20 |
| PGVNet (Ours) | **83.38** | – | – | **82.87** | – | – | – | 65.66 | 95.88 | **98.01** |

efficiency level. Compared with MinkNet, our method is more time-consuming, but our classification accuracy is higher, with a 5 % advantage.

Overall, our approach may fail to achieve an overwhelming performance in benchmarks as a whole compared with MinkNet because our method costs more time during data preprocessing, but it attains a more competitive performance than the mainstream methods and proves a better trade-off between accuracy and efficiency.

### 4.6. Ablation experiments

In our PGVNet, the AtEdgeConv module and attention fusion block play a vital role in roadside object recognition. To validate the effectiveness of these two modules, we conducted a series of ablation experiments on three models ($\text{PGVNet}_{\text{point}}$, $\text{PGVNet}_{\text{no-A}}$, and $\text{PGVNet}_{\text{1-A}}$) using different combinations of the components. Compared with our PGVNet, the $\text{PGVNet}_{\text{point}}$ only contains the point cloud branch listed in Section 3.1, mainly stacking two AtEdgeConv modules without the view feature extraction module and feature fusion module, which mainly extract local pointwise geometric features $\mathcal{F}_p$ for roadside object classification. $\text{PGVNet}_{\text{no-A}}$ removes all attention-fusion modules, and $\text{PGVNet}_{\text{1-A}}$ only stacks one attention-fusion module. The results of the ablation experiments are illustrated in Table 7 and Fig. 13.

Compared with the other point-based methods, $\text{PGVNet}_{\text{point}}$ achieved better performance with an OA of 88.21 % for the four categories and outperformed DGCNN by 1.2 %, which indicates the AtEdgeConv module's effectiveness. Similar to point-based methods, without the guidance of the global features from multiple views, $\text{PGVNet}_{\text{point}}$ also easily misclassified large objects as scenes in Fig. 13. Compared with $\text{PGVNet}_{\text{point}}$, $\text{PGVNet}_{\text{no-A}}$ improves the OA by 5.7 % and shows that the fusion of multiview and point cloud features helps to improve the distinguishability of the global descriptors of the roadside objects.

In the multimodal task, PGVNet achieves better performance than $\text{PGVNet}_{\text{no-A}}$ and $\text{PGVNet}_{\text{1-A}}$, especially in the car category. The F1-score of cars improves from 93.96 % to 96.49 %. We also observe that the proposed attention fusion module is beneficial for roadside object

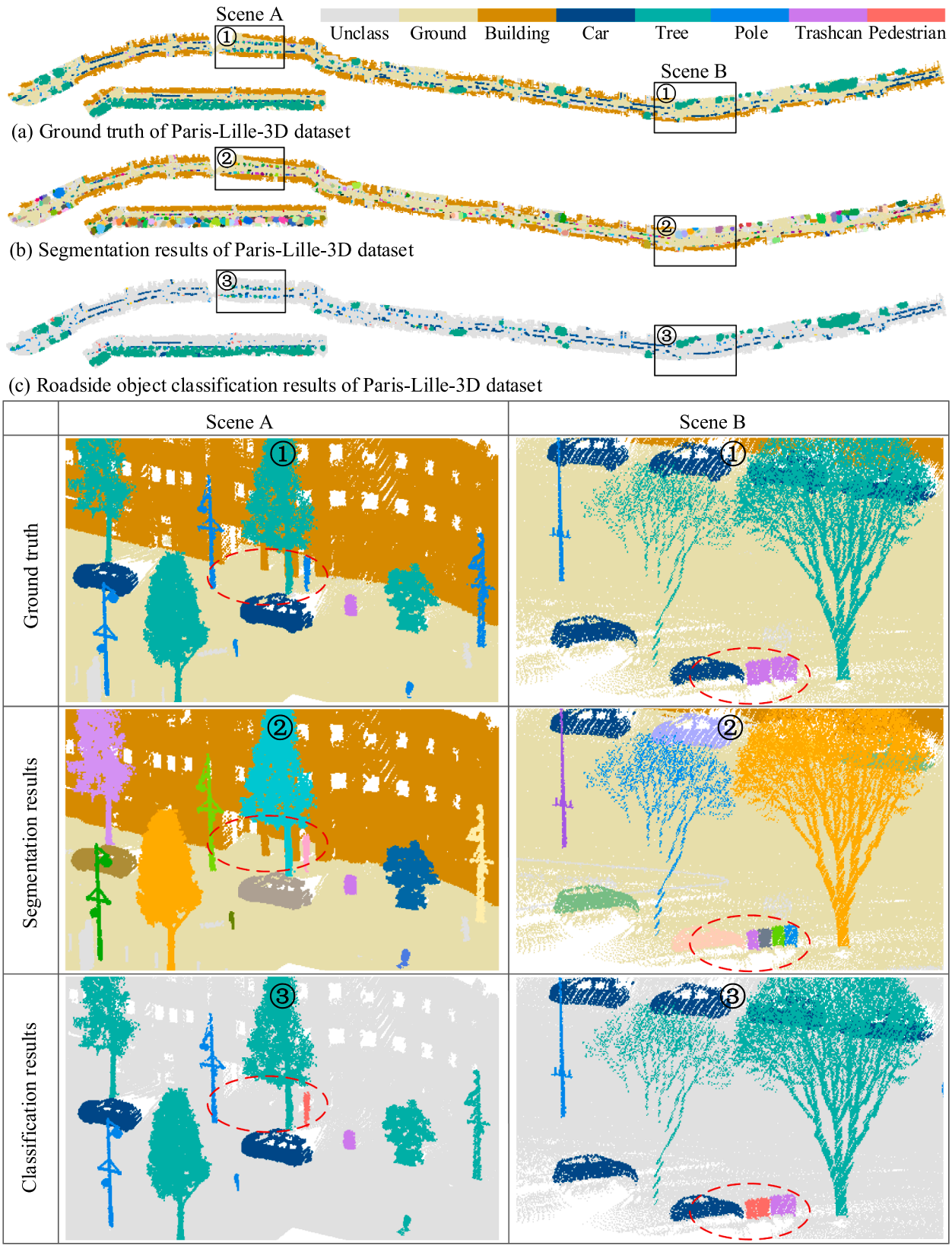



**Fig. 15.** The performance of our methods on the Paris-Lille-3D dataset: (a) ground truth (each class has a different color), (b) segmentation results (each roadside object has a color), and (c) classification results of our PGVNet. At the bottom, we illustrate two selected scenes. We circle two misclassified objects in red on the selected scenes. An incomplete pole (scene A) and an over-segmentation trashcan (scene B) are falsely detected as pedestrians due to the influence of over-segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

recognition. $PGVNet_{1\text{-}A}$, with one attention fusion module, outperforms $PGVNet_{no\text{-}A}$ without an attention fusion module by 1.1 % in OA. This is attributed to the attention fusion layers for exploring more discriminative features for roadside objects. Our attention fusion module makes it possible to use the global features of multiview models as soft attention masks to describe the correlations and significance of different geometric features from the point clouds.

Meanwhile, the visualization results of the attention coefficients in the attention ablation module further support our assumption. As shown in Fig. 14, the color changes from blue to red in the point cloud local features $\mathcal{F}_p$, and the attention fusion module coefficients indicate the features and weights of the different points. This shows that the distinguishable local features captured by the point cloud extraction module are mainly the edge features of an object, and the attention fusion module can improve them. However, Attention fusion block I only enhances a small range of features whose performance is not stable enough. In contrast, Attention fusion block II focuses on the more steady and quickly advanced structural properties of the roadside objects, such as the tree canopy and pole rods. In the PGVNet, two-layer attention fusion blocks are employed to improve the model's ability for critical feature extraction and the robustness of features. In this case, compared with other models, PGVNet achieves the best performance.

### 4.7. Influence of segmentation result on classification performance analyses

Since the degree of the incompleteness of roadside objects does affect the performance of our PGVNet, we explore how segmentation results affect the final classification result in this section. As our test datasets cover very large areas, if we manually label the pointwise ground truth, it will be very time consuming and hard to execute. To assess the segmentation method by Fang et al. (2020) quantitatively, we conduct extensive experiments on the training datasets of Paris-Lille-3D (NPM3D Benchmark Suite). The Paris-Lille-3D training dataset is a benchmark on point semantic segmentation with 10 coarse classes and consists of 2 km of MLS point clouds acquired by a Velodyne HDL-32e. Table 8 shows the segmentation results of our segmentation method. The IoU ($\text{IoU} = \frac{TP}{TP+FP+FN}$) scores of ground, building and other off-ground object class are 97.79 %, 97.16 % and 94.81 % respectively. After hierarchical segmentation, the roadside objects are partitioned into individual objects. This segmentation processing also achieved a promising performance and the IoU scores of the individual car, tree, pole and pedestrian are over 97.89 %, as illustrated in Table 8. After segmentation, we don't use this data to retrain the PGVNet and directly implement it to classify the roadside objects. Considering the effect of the segmentation operators, we calculate the final pointwise classification accuracy of PGVNet taking into account the segmentation influence and present them at the bottom of Table 9. In addition, we list all the per-class accuracy of the top-5 deep learning methods on the Paris-Lille-3D dataset in Table 9 for comparative analysis, including KP-Pyramind (Nie et al., 2022), PyramidPoint (Varney and Asari, 2022), FKAConv (Boulch et al., 2020), KPFCNN (Thomas et al., 2018), FG-Net(Liu et al., 2022). Compared with the top-5 semantic segmentation methods directly implementing deep learning networks on point clouds (without segmentation), our method gains competitive results with 83.38 % average IoU. Comparatively, our PGVNet works well on big-size roadside object recognition (e.g., tree and pole). As shown in Fig. 15, most cars and trees are correctly classified. Though some cars are under-segmented and lose the bottom heels, PGVNet overcome the influence of segmentation and works well on them. We observed that the result of the segmentation greatly impacts the final classification performance of PGVNet on small-size objects like poles and pedestrians. Compared with ground truth in Fig. 15, an incomplete pole (scene A) and an over-segmentation trashcan (scene B) in the selected scenes are falsely grouped into the pedestrian class.

## 5. Conclusion

This paper proposes a deep learning method, PGVNet, that combines point clouds and multiple views to recognize roadside objects from lidar point clouds. More discriminative and robust features are extracted to improve the method's performance in real scenarios. First, an attention-based edge convolution module is proposed to capture the local geometric features of the object point clouds. Second, a group fusion strategy is used to capture the optimal view feature. Third, an attention mechanism-based fusion module guides point cloud features to focus on the discriminative geometric structure by the view feature and generate global features for roadside object recognition. The evaluation indicates that the proposed method accurately recognizes most roadside cars, trees, traffic poles and small objects in complex real-world scenarios and achieves F1-score of 96.49 %, 97.70 %, 95.50 % and 89.45 % in test datasets, respectively. Remarkably, without retraining our model, the F1-score of cars, trees, and traffic poles in the ULS datasets are 81.55 %, 94.35 % and 97.78 %, respectively, demonstrating the broad range applicability of the proposed method to the data. Compared with the previous methods, the proposed method achieves promising gains in the car and small object category, with F1-score improvements of 5.21 % and 11.31 %, respectively. However, we note that some tree trunks are misclassified as traffic poles and mutilated cars are recognized as other objects, which means the result of the segmentation greatly impacts the final classification performance of our PGVNet on small-size objects like traffic poles and pedestrians. Further work will attempt to capture the discriminative features between similar objects or combine the image and point cloud for segmentation in real scenes.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Our study is jointly supported by the National Natural Science Foundation of China (NSFC) project (No. 42071446 and 41871380), and the Fujian Foreign Cooperation Project Foundation (No. 2020I0007). Thanks to Optech, StreetMapper, and Trimble Inc. for providing the datasets.

## References

Boulch, A., Puy, G., Marlet, R., 2020. FKAConv: Feature-kernel alignment for point cloud convolution. In: Proceedings of the Asian Conference on Computer Vision, pp. 381–399. https://doi.org/10.1007/978-3-030-69525-5_23.

Brock, A., Lim, T., Ritchie, J. M., Weston, N., 2016. Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236.

Che, E., Jung, J., Olsen, M.J., 2019. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. Sensors 19 (4), 810. https://www.mdpi.com/1424-8220/19/4/810.

Chen, C., Fragonara, L. Z., Tsourdos, A., 2019. GAPNet: Graph attention based point neural network for exploiting local feature of point cloud. arXiv preprint arXiv: 1905.08705.

Choy, C., Gwak, J., Savarese, S., 2019. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3075–3084. https://doi.org/10.1109/CVPR.2019.00319.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, pp. 3844-3852.

Dong, Z., Yang, B., Liu, Y., Liang, F., Li, B., Zang, Y., 2017. A novel binary shape context for 3D local surface description. ISPRS J. Photogramm. Remote Sens. 130, 431–452. https://doi.org/10.1016/j.isprsjprs.2017.06.012.

Fang, L., Shen, G., Luo, H., Chen, C., Zhao, Z., 2020. Automatic extraction of roadside traffic facilities from mobile laser scanning point clouds based on deep belief network. IEEE Trans. Intell. Transp. Syst. 22 (4), 1964–1980. https://doi.org/10.1109/TITS.2020.3017629.

Feng, Y., Zhang, Z., Zhao, X., Ji, R., Gao, Y., 2018. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition, pp. 264–272. https://doi.org/10.1109/CVPR.2018.00035.
Guo, H., Wang, J., Gao, Y., Li, J., Lu, H., 2016. Multi-view 3D object retrieval with deep embedding network. IEEE Trans. Image Process. 25 (12), 5526–5537. https://doi.org/10.1109/TIP.2016.260 9814.
Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 43 (12), 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434.
Han, X., Dong, Z., Yang, B.J.I.J.O.P., Sensing, R., 2021. A point-based deep learning network for semantic segmentation of MLS point clouds. 175, 199-214. https://doi.org/10.1016/j.isprsjprs.2021.03.001.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
Klokov, R., Lempitsky, V., 2017. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 863–872. https://doi.org/10.1109/ICCV.2017.99.
Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp, 1097–1105.
Lan, S., Yu, R., Yu, G., Davis, L.S., 2019. Modeling local geometric structure of 3d point clouds using geo-cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 998–1008. https://doi.org/10.1109/CVPR.2019.00109.
Lehtomäki, M., Jaakkola, A., Hyyppä, J., Lampinen, J., Kaartinen, H., Kukko, A., Hyyppä, H., 2015. Object classification and recognition from mobile laser scanning point clouds in a road environment. IEEE Trans. Geosci. Remote Sens. 54 (2), 1226–1239. https://doi.org/10.1109/TGRS.2015.2476502.
Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems, pp. 820–830.
Li, W., Luo, Z., Xiao, Z., Chen, Y., Wang, C., Li, J., 2021. A GCN-based method for extracting power lines and pylons from airborne LiDAR data. IEEE Trans. Geosci. Remote Sens. https://doi.org/10.1109/TGRS.2021.3076107.
Liu, K., Gao, Z., Lin, F., Chen, B.M., 2022. FG-Net: A Fast and Accurate Framework for Large-Scale LiDAR Point Cloud Understanding. IEEE Transactions on. Cybernetics. https://doi.org/10.1109/TCYB.2022.3159815.
Luo, Z., Li, J., Xiao, Z., Mou, Z.G., Cai, X., Wang, C., 2019. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. ISPRS J. Photogramm. Remote Sens. 150, 44–58. https://doi.org/10.1016/j.isprsjprs.2019.01.024.
Maturana, D., Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 922–928. https://doi.org/10.1109/IROS.2015.7353481.
Meyer, G.P., Charland, J., Hegde, D., Laddha, A., Vallespi-Gonzalez, C., 2019. Sensor fusion for joint 3d object detection and semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp. 1230–1237. https://doi.org/10.1109/CVPRW.2019.00162.
Mi, X., Yang, B., Dong, Z., Chen, C., Gu, J., 2021a. Automated 3D Road Boundary Extraction and Vectorization Using MLS Point Clouds. IEEE Trans. Intell. Transp. Syst. 23 (6), 5287–5297. https://doi.org/10.1109/TITS.2021.3052882.
Mi, X., Yang, B., Dong, Z., Liu, C., Zong, Z., Yuan, Z., 2021b. A two-stage approach for road marking extraction and modeling using MLS point clouds. ISPRS J. Photogramm. Remote Sens. 180, 255–268. https://doi.org/10.1016/j.isprsjprs.2021.07.012.
Nie, D., Lan, R., Wang, L., Ren, X., 2022. Pyramid Architecture for Multi-Scale Processing in Point Cloud Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17284–17294.
Nie, W., Zhao, Y., Song, D., Gao, Y., 2021. DAN: Deep-attention network for 3D shape recognition. IEEE Trans. Image Process. 30, 4371–4383. https://doi.org/10.1109/TIP.2021.3071687.
Poux, F., Billen, R., 2019. Voxel-based 3D point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs deep learning methods. ISPRS Int. J. Geo-Inf. 8 (5), 213. https://doi.org/10.3390/ijgi8050213.
Poux, F., Mattes, C., Selman, Z., Kobbelt, L., 2022. Automatic region-growing system for the segmentation of large point clouds. Autom. Constr. 138, 104250. https://doi.org/10.1016/j.autcon.2022.104250.
Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. PointNet++: Deep hierarchical feature learning on Point Sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5099–5108.
Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660. https://doi.org/10.1109/CVPR.2017.16.
Riegler, G., Osman Ulusoy, A., Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3577–3586. https://doi.org/10.1109/CVPR.2017.701.
Roynard, X., Deschaud, J.E., Goulette, F., 2018b. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. Int. J. Robot. Res. 37 (6), 545–557. https://doi.org/10.1177/0278364918767506.
Roynard, X., Deschaud, J.E., Goulette, F., 2018. Classification of point cloud scenes with multiscale voxel deep network. arXiv preprint arXiv:1804.03583.
Shi, W., Rajkumar, R., 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In: ProceedIngs of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1711–1719.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 945–953. https://doi.org/10.1109/ICCV.2015.114.
Thomas, H., Goulette, F., Deschaud, J.-E., Marcotegui, B., LeGall, Y., 2018. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In: 2018 International Conference on 3D Vision (3DV). IEEE, pp. 390–398. https://doi.org/10.1109/3DV.2018.00052.
Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6411–6420. https://doi.org/10.1109/ICCV.2019.00651.
Varney, N., Asari, V.K., 2022. Pyramid point: A multi-level focusing network for revisiting feature layers. IEEE Geosci. Remote Sens. Lett. https://doi.org/10.1109/LGRS.2022.3191743.
Vishwanath, K.V., Gupta, D., Vahdat, A., Yocum, K., 2009. Modelnet: Towards a datacenter emulation environment. In: 2009 IEEE Ninth International Conference on Peer-to-Peer Computing. IEEE, pp. 81–82. https://doi.org/10.1109/P2P.2009.5284497.
Wang, L., Ouyang, W., Wang, X., Lu, H., 2015. Visual tracking with fully convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3119–3127. https://doi.org/10.1109/ICCV.2015.357.
Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019a. Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10296–10305.
Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019b. Dynamic graph cnn for learning on point clouds. ACM Trans. Graphics 38 (5), 1–12. https://doi.org/10.1145/3326362.
Wen, C., Li, X., Yao, X., Peng, L., Chi, T., 2021. Airborne LiDAR point cloud classification with global-local graph attention convolution neural network. ISPRS J. Photogramm. Remote Sens. 173, 181–194. https://doi.org/10.1016/j.isprsjprs.2021.01.007.
Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9621–9630.
Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920.
Xiao, W., Vallet, B., Schindler, K., Paparoditis, N., 2016. Street-side vehicle detection, classification and change detection using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 114, 166–178. https://doi.org/10.1016/j.isprsjprs.2016.02.007.
Xu, D., Anguelov, D., Jain, A., 2018. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 244–253.
Xu, Y., Zheng, C., Xu, R., Quan, Y., Ling, H., 2021. Multi-View 3D shape recognition via correspondence-aware deep learning. IEEE Trans. Image Process. 30, 5299–5312. https://doi.org/10.1109/TI P. 2021.3082310.
Yan, L., Li, Z., Liu, H., Tan, J., Zhao, S., Chen, C., 2017. Detection and classification of pole-like road objects from mobile LiDAR data in motorway environment. Opt. Laser Technol. 97, 272–283. https://doi.org/10.1016/j.optlastec.2017.06.015.
Yang, B., Dong, Z., Liu, Y., Liang, F., Wang, Y., 2017. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 126, 180–194. https://doi.org/10.1016/j.isprsjprs.2017.02.014.
Yang, Z., Wang, L., 2019. Learning relationships for multi-view 3D object recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7505–7514.
Ye, M., Xu, S., Cao, T., 2020. Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1631–1640.
You, H., Feng, Y., Ji, R., Gao, Y., 2018. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 1310–1318. https://doi.org/10.1145/3240508.3240702.
You, H., Feng, Y., Zhao, X., Zou, C., Ji, R., Gao, Y., 2019. PVRNet: Point-view relation neural network for 3D shape recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9119–9126. https://doi.org/10.1609/aaai.v33i01.33019119.
Zhang, Y., Rabbat, M., 2018. A graph-cnn for 3d point cloud classification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 6279–6283. https://doi.org/10.1109/ICASSP.2018.8462291.
Zhao, H., Jiang, L., Fu, C.-W., Jia, J., 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5565–5573.