applied sciences

MDPI

Article

Semantic Segmentation of 3D Point Clouds in Outdoor Environments Based on Local Dual-Enhancement

Kai Zhang [1], Yi An [1,2,*], Yunhao Cui [3] and Hongxiang Dong [1]

[1] School of Electrical Engineering, Xinjiang University, Urumqi 830046, China; 107552101490@stu.xju.edu.cn (K.Z.); 107552104010@stu.xju.edu.cn (H.D.)
[2] School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China
[3] School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471023, China; 9906412@haust.edu.cn
* Correspondence: anyi@dlut.edu.cn

Abstract: Semantic segmentation of 3D point clouds in drivable areas is very important for unmanned vehicles. Due to the imbalance between the size of various outdoor scene objects and the sample size, the object boundaries are not clear, and small sample features cannot be extracted. As a result, the semantic segmentation accuracy of 3D point clouds in outdoor environment is not high. To solve these problems, we propose a local dual-enhancement network (LDE-Net) for semantic segmentation of 3D point clouds in outdoor environments for unmanned vehicles. The network is composed of local-global feature extraction modules, and a local feature aggregation classifier. The local-global feature extraction module captures both local and global features, which can improve the accuracy and robustness of semantic segmentation. The local feature aggregation classifier considers the feature information of neighboring points to ensure clarity of object boundaries and the high overall accuracy of semantic segmentation. Experimental results show that provides clearer boundaries between various objects, and has higher identification accuracy for small sample objects. The LDE-Net has good performance for semantic segmentation of 3D point clouds in outdoor environments.

Keywords: 3D point clouds; local augmentation; semantic segmentation; outdoor environment; unmanned vehicles

Citation: Zhang, K.; An, Y.; Cui, Y.; Dong, H. Semantic Segmentation of 3D Point Clouds in Outdoor Environments Based on Local Dual-Enhancement. *Appl. Sci.* **2024**, *14*, 1777. https://doi.org/10.3390/app14051777

Academic Editors: João M. F. Rodrigues

Received: 7 January 2024
Revised: 1 February 2024
Accepted: 19 February 2024
Published: 22 February 2024

1. Introduction

With the rapid development of mobile robot technology and artificial intelligence, ground unmanned vehicles have been widely used in various fields such as logistics, transportation, security, inspection, and so on [1]. Driverless technology can improve production efficiency, reduce safety risks, reduce operating costs, increase resource utilization, and promote industrial transformation and upgrading [2].

Meanwhile, the traditional automotive industry, relying on the development of artificial intelligence, is vigorously researching and developing autonomous driving technology. The outstanding role of autonomous driving technology in reducing driver intensity and improving driving safety has given this technology a promising development prospect.

The core technical system of autonomous driving can be mainly divided into three levels: perception, decision-making, and execution. Perceiving and locating the surrounding environment is a prerequisite for autonomous driving technology.

Currently, there are roughly two approaches to autonomous driving perception technology: one is a machine vision-centric solution with millimeter-wave radar and cameras, typically represented by companies such as Tesla, Mobileye, Baidu Apollo, etc.; the other is a sensor route centered on high-precision maps and LiDAR, represented by companies such as Waymo and Huawei.

In the debate between pure vision and LiDAR, some argue that pure vision solutions are sufficient for most road scenarios, while LiDAR is merely a complement. They believe

that human drivers primarily rely on visual information during driving, making pure vision solutions more aligned with human driving habits. Furthermore, with the continuous development of computer vision technology, the accuracy and reliability of pure vision solutions are constantly improving. However, others contend that LiDAR is crucial for autonomous vehicles. They argue that in complex road scenarios and adverse weather conditions, pure vision solutions struggle to ensure accuracy, whereas LiDAR can provide more reliable information. Additionally, LiDAR can be fused with other sensors to further enhance the accuracy and safety of autonomous vehicles.

Previously, pure vision solutions were held as overwhelmingly dominant. However, with the continuous advancement of LiDAR technology, an increasing number of vehicles are being equipped with LiDAR. Furthermore, the declining cost of LiDAR will further promote its development. Therefore, processing the data collected by LiDAR has become increasingly important.

The 3D point clouds obtained from LIDAR scanning are represented as a set of points. The significance of semantic segmentation lies in assigning labels to each point, allowing for differentiation of the entire point clouds and achieving the purpose of initial visual understanding. With semantically calibrated 3D point clouds, unmanned vehicles can accurately identify actionable areas, enabling better decision-making and control. It can also extract slope surfaces for safe slope calculations to prevent accidents. Therefore, accurate semantic segmentation of 3D point clouds is a crucial prerequisite for constructing driverless technology, and its precision determines the reliability of autonomous driving, the accuracy of safety inspections, and the precision of 3D modeling.

In recent years, many deep learning methods have been applied to 3D point cloud semantic segmentation [3,4]. These methods can be roughly classified into three categories: projection-based methods, voxel-based methods, and direct point-based methods. Projection-based methods project the 3D point clouds onto a plane and turn them into 2D images, then use 2D images for semantic segmentation, which lose the spatial characteristics of the original 3D point clouds. Voxel-based methods transform 3D point clouds into dense or sparse 3D grids. Semantic segmentation is performed using standard 3D convolution. Computing the dense grids requires significant computational resources, making it impractical for real-world applications. Transforming into sparse grids may result in the loss of certain 3D features. In contrast, point-based methods directly operate on the 3D point clouds, and can balance the semantic segmentation effect and computational cost. Recently, scholars have proposed many methods for semantic segmentation of scenes, such as RandLA-Net [5].

Due to the enormous scale of outdoor scenes, extremely uneven distribution of data samples, and high requirements for clarity of object boundaries, existing methods were not specifically designed for outdoor environments and thus cannot perform well in such environments [6]. Training deep learning network models for semantic segmentation typically utilizes supervised learning, which requires a large number of outdoor sites and cloud semantic segmentation data sets. Currently, there are no publicly available datasets specifically designed for outdoor environments that we can use directly. The problems we have encountered are summarized as follows: the mainstream semantic segmentation methods do not exhibit high accuracy, there is inadequate extraction of mine boundaries' features, and small-scale data features are also inadequately extracted.

To solve the above problems, we designed LDE-Net, a semantic segmentation network of 3D point clouds in outdoor environments for driverless unmanned vehicles. As shown in Figure 1, the LDE-Net consists of an encoder-decoder framework, local-global feature extraction modules, and a local feature aggregation classifier. The LDE-Net utilizes the local-global information extraction module, effectively resolving the issue of disconnected neighborhoods among 3D point clouds in outdoor environments. The local aggregation classifier can consider neighborhood information at the final classification stage to ensure more accurate classification in outdoor environments.

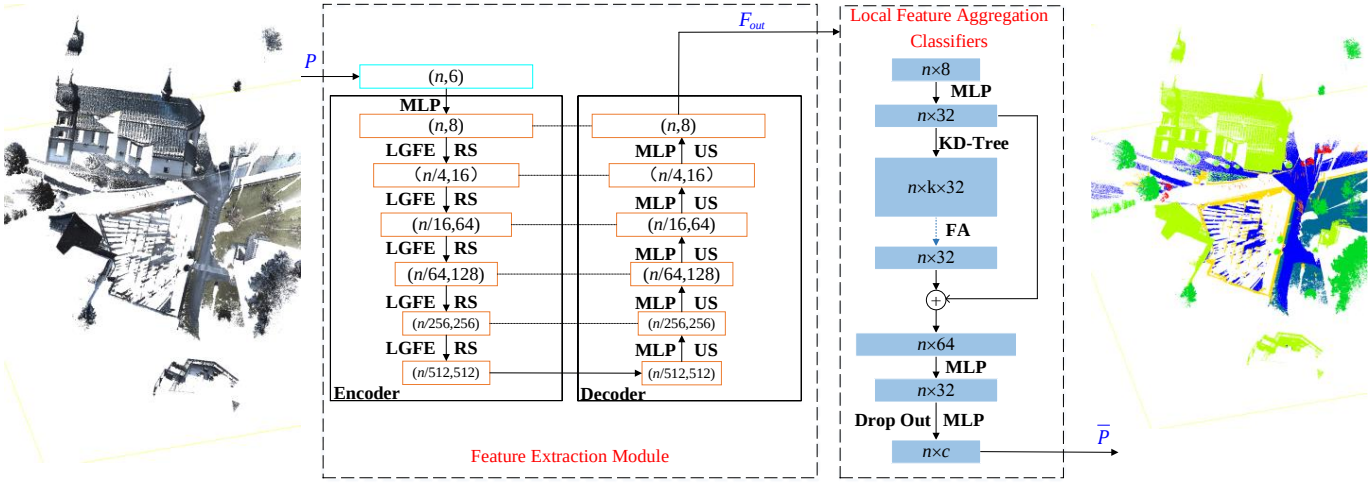

Figure 1. Framework of the LDE-Net.

The rest of this article is organized as follows: Section 2 reviews related work that has been carried out; Section 3 provides a detailed introduction to the LDE-Net; Section 4 presents experimental results to demonstrate the advantages of the network; and Section 5 draws conclusions.

2. Related Work

In this section, we will review the research related to dataset and 3D point cloud semantic segmentation.

In recent years, due to the rapid development of deep learning, a variety of data sets applied to visual understanding have appeared, thus promoting the development of visual understanding. Most datasets utilize RGB cameras to capture 2D images with color features, for example [6–10], including images captured in different weather and lighting conditions. However, Geiger, A. [11] has pioneered the provision of a multi-modal dataset KITTI, which provides dense point clouds, frontal stereo images, and GPS/IMU data from LiDAR sensors. Meanwhile, Choi, Y. [12] also provides data composed of RGB and thermal cameras, RGB stereo sound, 3D LiDAR, and GPS/IMU. Chen, Y. [13] has done the same work as well. Caesar, H. [14] introduces the multi-view mode 3D detection dataset Nuscenes. Huang, X. [15] and Xibin Song, X. [16] focus on creating datasets for pixel-level semantic segmentation tasks, including scene parsing, 3D car instance, and lane segmentation tasks. Haibao Yu [17] focuses on 3D detection tasks using LiDAR.

The existing methods for extracting features from 3D point clouds can generally be divided into three groups: projection-based [18–21], voxel-based [22–24], and point-based [25–29]. Table 1 shows the relevant details.

Table 1. Comparison of semantic segmentation methods for 3D point clouds.

	Methods	Dataset Used	Characteristic	Limited
Projection-based	SnapNet [18]	Semantic3D	These methods project a 3D point cloud onto a 2D image plane. They can be segmented by using the mature 2D image processing technology. They have low computational complexity.	These methods lose some information in the process of projection. They have poor segmentation for complex 3D shapes. They select the appropriate projection angle and parameters manually.
	SqueezeSeg [19]	KITTI		
	SqueezeSegV2 [20]	KITTI		
	RangeNet++ [21]	KITTI		

Table 1. *Cont.*

	Methods	Dataset Used	Characteristic	Limited
Voxel-based	SPLATNet [22]	RueMonge2014, ShapeNet	These methods convert the point cloud data into a voxel grid. They use voxel information to represent 3D objects. They can retain neighborhood information.	These methods may result in information loss. They have high time and space complexity. They can be difficult to select the correct voxel resolution.
	FCPN [23]	ScanNets, ShapeNet		
	SSCNs [24]	ShapeNet, NYU Depth (v2)		
Point-based	PointNet [25]	ModelNet40, ShapeNet, Stanford 3D	These methods directly process 3D point cloud data and preserve geometric information and details in the point cloud. They can effectively deal with the sparsity of the point cloud. They have low computational and memory consumption.	Most of these methods adopt expensive neighborhood search mechanisms. They require large scale, high quality annotated data. They are computationally complex.
	PointSIEF [26]	S3DIS, ScanNet		
	PointWeb [28]	S3DIS, ScanNet, ModelNet40		
	ShellNet [29]	ModelNet40, ShapeNet, ScanNet, S3DIS, Semantic3D		
	RandLA-Net [5]	SemanticKITTI, Semantic3D, S3DIS		

Projection-based methods first project the 3D point clouds onto a 2D plane, then use 2D convolution to obtain the semantic labels of each pixel, and finally fuse the multi-view semantic labels to obtain the semantic labels of each point. Lawin et al. [18] first projected a 3D point cloud onto 2D planes from multiple virtual camera views. To achieve fast and accurate segmentation of 3D point clouds, Wu et al. [19,20] proposed an end-to-end network based on SqueezeNet and SqueezeNetV2. Milioto et al. [21] proposed RangeNet++ for real-time semantic segmentation of LiDAR point clouds.

Voxel-based methods voxelize the point clouds and then perform semantic segmentation using standard 3D convolution. Jing and Suya [22] first convert a point cloud into a group of voxels. Then, they input these data into 3D CNN for voxel segmentation. Finally, they assign all points within a voxel with the same label as the voxel. Rethage et al. [23] proposed a fully convolutional network (FCN) which extracts different levels of geometric relationships layer by layer from the point clouds using 3D convolution and weighted average. The FCN is used for feature extraction and long-range dependency integration, and it can effectively handle colored point clouds. Graham et al. [24] proposed a deep learning network based on index structures that can significantly reduce both memory and computational costs.

Point-based methods directly take the 3D point clouds into the network. Charles et al. [25] designed a shared MLP to learn the features of each point. Jiang et al. [26] implemented orientation encoding and scale awareness using a three-stage and order-wise convolution method. This method effectively stacks and encodes information from eight spatial directions, which concatenates multi-scale features for adaptive processing of different scales. Engelmann et al. [27] developed an approach that differs from the grouping technique used in PointNet++. They defined two neighborhoods in both world space and feature space by using K-means clustering and K-nearest neighbors (KNN). Additionally, they introduced pairwise distance loss and centroid loss to better regularize feature learning. To simulate interactions between different points, PointWeb investigates the relationships between all point pairs within a local region by densely constructing locally fully connected networks [28]. They proposed an adaptive feature adjustment module for information exchange and feature refinement, which aids in learning discriminative feature representations. PointWeb can effectively extract the local features of the point cloud, but it processes the point cloud in a grid-based way. Therefore, PointWeb does not work well when dealing with irregular point cloud shapes. Zhang et al. [29] presented a permutation-invariant convolution method based on the statistical quantities of concentric spherical shells. This method queries a set of multiscale concentric spheres, summarizes the statistical

data using max-pooling operations in different shells, and obtains the final convolution output using MLP and 1D convolution. However, ShellNet's processing efficiency for large-scale point cloud data is low. The number and radius of shells need to be manually set, which may affect their generalization ability.

The above methods exhibit lower accuracy in semantic segmentation of large-scale 3D point cloud data in urban environments. Hu et al. [5] proposed an efficient and lightweight RandLA-Net that can be used for colored point cloud segmentation. This network utilizes random point sampling to achieve high efficiency in terms of memory and computation, and further proposes a local feature aggregation module to capture and preserve geometric features.

3. Semantic Segmentation of 3D Point Clouds

This article proposes a novel semantic segmentation network of 3D point clouds in outdoor environments for driverless vehicles. As shown in Figure 1, the LDE-Net includes a feature extraction network and a local feature aggregation classifier. The feature extraction network consists of an encoder and a decoder, with local-global feature extraction modules (LGFE) embedded in the encoder.

3.1. Framework of the LDE-Net

The 3D point cloud $P = \{\boldsymbol{p_i} = (x_i, y_i, z_i) \mid 1 \leq i \leq n\}$ is inputted into the feature extraction network, which then outputs the 3D point cloud $\overline{P} = \{\overline{\boldsymbol{p}}_{\boldsymbol{i}} = (x_i, y_i, z_i, l_i) \mid 1 \leq i \leq n\}$ with semantic labels. n is the total number of points, (x_i, y_i, z_i) is the coordinate of a point $\boldsymbol{p_i}$, and l_i is the semantic category information of a point $\boldsymbol{p_i}$. The feature extraction network can extract local and global features for each point of the input outdoor 3D point cloud. Local-global feature extraction modules can better classify each point according to the extracted features.

The feature extraction network processes the 3D point cloud P and obtains the output feature $F_{out} = \left\{\boldsymbol{f_i^{out}} = \left(f_i^1, f_i^2, f_i^3, f_i^4, f_i^5, f_i^6, f_i^7, f_i^8\right) \middle| 1 \leq i \leq n\right\}$. The output feature F_{out} is classified by the local feature aggregation classifier to finally obtain the 3D point cloud $\overline{P}$ with semantic labels.

3.2. Feature Extraction Network

The 3D point cloud P is fed into the feature extraction network. Our challenge is to design a feature extraction network that is tailored to the specific structure of 3D point clouds in order to extract features and achieve a more accurate semantic segmentation of point clouds. The feature extraction network is composed of an encoder and a decoder.

3.2.1. Encoder-Decoder Module

As shown in Figure 1, the input of the encoder module is the 3D point cloud P. It contains spatial information and color information of the 3D point cloud P. The 3D point cloud P is extracted by a multilayer perceptron network and the dimension is unified to 8. The encoder module reduces the number of points by random sampling and learns the spatial contextual features of each point through the local-global feature extraction module (LGFE) five times. The number of points is gradually decreased from n to $n/512$, while the feature dimension is learned from 8 to 512.

In the decoder module, the features are up-sampled by nearest-neighbor interpolation and further condensed through the multilayer perceptron (MLP) network. The total number of points is restored to n, and the dimension of the features is condensed to 8. Finally, we can obtain the output feature F_{out} of the feature extraction network.

3.2.2. The Local-Global Feature Extraction Module

The local-global feature extraction module is embedded in the encoder module. As shown in Figure 2, the input of the module includes the spatial information, color information and feature information F_e learned previously. After feature learning by the

local-global feature extraction module, the extracted features F_o is outputted. The local information is extracted by using the dual distance feature extraction module, while the global information is extracted using by the global feature extraction module.

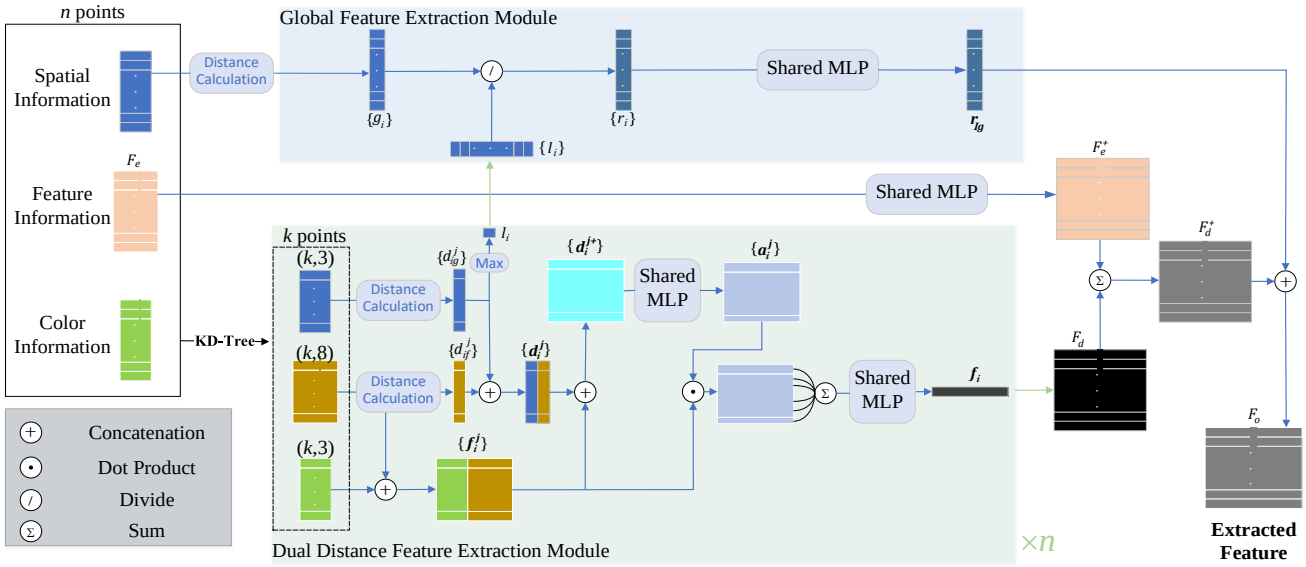

Figure 2. The local-global feature extraction module (LGFE).

The dual distance feature extraction module From a 3D perspective, distance is an important indicator of the correlation between points. The smaller the distance between points, the higher their correlation. Distance includes not only the spatial distance between two points but also the feature distance between two points. Therefore, we propose the dual distance feature extraction module based on the spatial and feature distances between two points, which can automatically learn the effective local information features. Its specific structure is shown in Figure 2.

The input of the dual distance feature extraction module is the spatial information, feature information, and color information of the point $\boldsymbol{p_i}$ and its k neighboring points. The dimension of spatial information and color vector information is $(k, 3)$, and the dimension of input feature is $(k, 8)$.

In this module, we calculate the spatial distance d_{ig}^j and feature distance d_{if}^j between the point $\boldsymbol{p_i}$ and its j-th neighboring point $\boldsymbol{p_i^j}$ as

$$\begin{aligned} d_{ig}^j &= \left|\boldsymbol{p_i} - \boldsymbol{p_i^j}\right| \quad j = 1, \cdots, k \\ d_{if}^j &= \left|f(\boldsymbol{p_i}) - f\left(\boldsymbol{p_i^j}\right)\right| \quad j = 1, \cdots, k \end{aligned} \tag{1}$$

where $|\cdot|$ is the L_1 norm. $f(\boldsymbol{p_i})$ is the i-th point feature and $f\left(\boldsymbol{p_i^j}\right)$ is its j-th neighboring point feature. Since the features are automatically learned by the network, we use the negative exponent of both distances to learn two attention concentration weights and use λ to adjust d_{if}^j to address instability to obtain the dual distance $\boldsymbol{d_i^j}$.

$$\boldsymbol{d_i^j} = exp\left(-d_{ig}^j\right) \oplus \lambda exp\left(-d_{if}^j\right) \tag{2}$$

where $\oplus$ is the concatenation operator.

Then, the dual distance $\boldsymbol{d_i^j}$ and the feature $\boldsymbol{f_i^j}$ are concatenated as

$$\boldsymbol{d_i^{j+}} = \boldsymbol{d_i^j} \oplus \boldsymbol{f_i^j} \tag{3}$$

where $\boldsymbol{f_i^j}$ is obtained by concatenating the color information and the feature information of the k neighboring points of the point $\boldsymbol{p_i}$.

And a shared MLP and softmax are applied to $\boldsymbol{d_i^{j+}}$ to obtain $\boldsymbol{a_i^j}$

$$\boldsymbol{a_i^j} = \text{softmax}\left(\text{MLP}\left(\boldsymbol{d_i^{j+}}\right)\right). \tag{4}$$

Finally, the output feature of the dual distance feature extraction module is obtained by

$$\boldsymbol{f_i} = \text{MLP}\left(\sum_{j=1}^{k}\left(\boldsymbol{a_i^j} \odot \boldsymbol{f_i^j}\right)\right) \tag{5}$$

Global Feature Extraction Module The dual distance feature extraction module can extract the semantic features between two points within the neighborhood, but its power to judge the semantic segmentation of the overall 3D point clouds is insufficient. Since the semantic segmentation of the entire 3D point clouds requires integration and global information, we propose a global feature extraction module to learn global features from the 3D point cloud P.

As shown in Figure 3, we use 2D point clouds to express the relationship between the local region and the global region in the 3D point clouds. We use the farthest distance l_i between a given point $\boldsymbol{p_i}$ and the local boundary, and the farthest distance g_i between the point $\boldsymbol{p_i}$ and the global boundary to extract the global features.

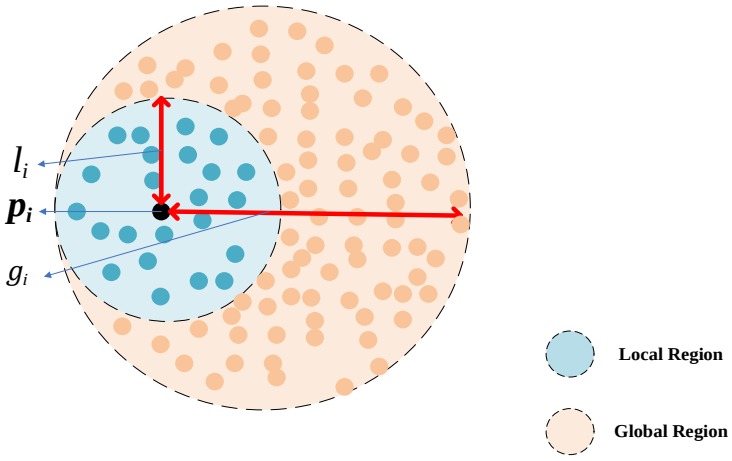

Figure 3. Global feature extraction.

Then, the global feature r_i of the point $\boldsymbol{p_i}$ is computed as

$$r_i = \frac{exp(-l_i)}{exp(-g_i)}. \tag{6}$$

Furthermore, MLP is used to further extract the global information

$$\boldsymbol{r_{ig}} = \text{MLP}(\ \{r_{\ i} \mid 1 \leq i \leq n\}). \tag{7}$$

After both the local feature and the global feature are extracted, we combine them into complete extracted features. The dual distance feature extraction module can calculate and extract the neighborhood features of each point. When we extract the neighborhood features of n points, we stitch them together to get $F_d = \{\boldsymbol{f_i}\}_{i=1}^{n}$.

Eventually, the local features of all points converge to F_d^+

$$F_d^+ = F_d + \text{MLP}(F_e). \tag{8}$$

After the local-global feature extraction module, the extracted feature F_o of the local-global feature extraction module are obtained.

$$F_o = \boldsymbol{r_{ig}} \oplus F_d^+. \tag{9}$$

3.3. Local Feature Aggregation Classifiers

The semantic features F_{out} extracted by the Feature Extraction network are transformed into specific semantic labels by the classifier. In previous semantic segmentation of 3D point clouds, the classifier generates point-wise semantic labels individually via MLPs implemented by fully connected layers. A fully connected layer consists of a linear transformation and a non-linear activation function. However, the non-linear activation function results in neighbor inconsistency in the prediction. To address this problem, we introduce a feature aggregation (FA) module into the MLP classifier, which enables the classifier to reference neighborhood information during classification. With this module, the contextual awareness ability of classification is enhanced.

For a given point $\boldsymbol{p_i}$ in P, we have used the KD-Tree algorithm to construct its neighborhood $N_i = \left\{ \boldsymbol{p_i^j} \middle| 1 \le j \le k \right\}$.

As shown in Figure 4, the feature aggregation module searches for the k nearest neighboring points and represents the feature of the neighboring point $\boldsymbol{p_i^j}$ as $\boldsymbol{e_i^j}$, which comes from the output feature F_{out} that has been processed by MLP.

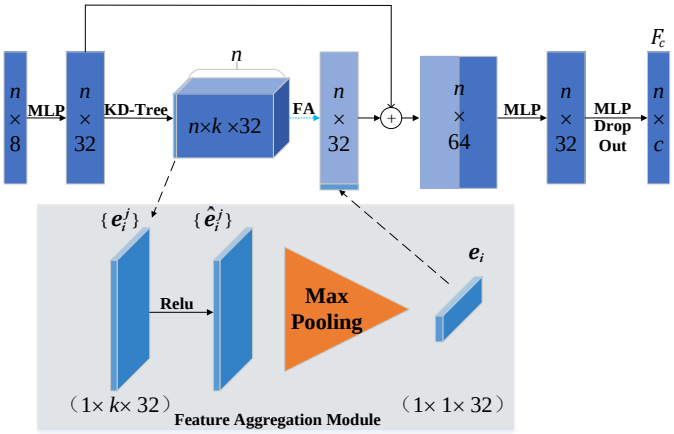

Figure 4. Local feature aggregation classifier.

The feature $\boldsymbol{e_i^j}$ is further addressed by the learnable weight w_i and activation function to obtain the neighborhood feature:

$$\hat{\boldsymbol{e}}_i^j = \mathrm{ReLU}\left(w_i \boldsymbol{e_i^j}\right). \tag{10}$$

Finally, we use channel-wise max pooling to aggregate the neighborhood feature $\hat{\boldsymbol{e}}_i^j$ to obtain the final feature of the point $\boldsymbol{p_i}$:

$$\boldsymbol{e_i} = \mathrm{maxpooling}(\{\hat{\boldsymbol{e}}_i^j \middle| 1 \le j \le k\}). \tag{11}$$

The max pooling operation may cause the disappearance of the feature of the origin point. To address this issue, we combine the original feature with the feature after max pooling. Finally, the final output feature F_c are obtained after MLPs and dropout. F_c is represented by a 2D matrix $(n \times c)$, where c represents the number of categories in the entire dataset.

4. Experiment and Analysis

In this section, we evaluate our LDE-Net on two typical point cloud in the outdoor environment dataset: Semantic3D and SemanticKITTI.

4.1. Dataset Introduction

Semantic3D is a large outdoor point cloud dataset with more than 3 billion points from the real world, including urban and rural scenarios. It consists of 15 training point clouds and 15 online test point clouds. In addition to coordinate and color information, each point also has an intensity value, but we do not use them. Each point is annotated with one of the semantic tags from the eight classes.

SemanticKITTI consists of 43,552 densely annotated LIDAR scans belonging to 21 sequences. Each scan is a large point cloud with ~105 points spanning up to $160 \times 160 \times 20$ m^3 in 3D space. Officially, sequences 00 to 07 and 09 to 10 (19,130 scans) are used for training, sequence 08 (4071 scans) for validation, and sequence 11 to 21 (20,351 scans) for online testing. The original 3D points only have 3D coordinates and no color information. The mIoU scores in more than 19 categories are used as standard indicators.

4.2. Experimental Detail

We deploy the LDE-Net network on a Linux server with a hardware environment, which consists of an Nvidia V100 GPU with 24 GB memory and 72 GB RAM. The software environment includes Cuda version 11.0, Cudnn version 8.74, TensorFlow version 1.15, and Python version 3.8.

In the model training process, we set the batch size to 8 and used an Adam optimizer with an initial learning rate of 0.01. We trained the model for 100 epochs, with a learning rate reduction of 5% after each epoch. The number of nearest neighbor points searched by KNN was set to 16. A fixed number of points (40,960 points) were sampled from each training point cloud and fed into the network for training. Multiple non-repetitive samplings were performed to ensure that the network could learn the features of the original point cloud.

During the testing phase, we sampled a fixed number of points from each point cloud as input for inference. To ensure that all points in each point cloud were inferred, we performed non-repetitive sampling multiple times until all points were inferred to complete the point cloud test. In order to evaluate the performance of the model, we used the following evaluation methods for comprehensive evaluation: overall accuracy (OA), mean class accuracy (mAcc), and mean intersection over union accuracy (mIoU). These can be defined as follows:

$$\mathrm{OA} = \frac{\sum_{i=0}^{n-1} p_{ii}}{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij}} \tag{12}$$

$$\mathrm{IoU} = \frac{p_{ii}}{\sum_{j=0}^{n-1} p_{ij} + \sum_{j=0}^{n-1} p_{ji} - p_{ii}} \tag{13}$$

$$\mathrm{mIoU} = \frac{1}{n} \sum_{i=0}^{n-1} \frac{p_{ii}}{\sum_{j=0}^{n-1} p_{ij} + \sum_{j=0}^{n-1} p_{ji} - p_{ii}} \tag{14}$$

where p_{ij} represents the number of points with ground truth label i and predicted label j in the point clouds and n represents the number of semantic classes.

4.3. Semantic3D

Table 2 reproduces the experimental results of several classic algorithms for semantic segmentation of 3D point clouds in Semantic3D dataset, including SnapNet_ [17], RF_MSSF [30], SEGCloud [31], ShellNet [28], GAENet [32], SPG [33], KPConv [34], and RandLA-Net [5]. In the projection-based approach, SnapNet_ projects 3D point clouds into multi-view images and then extracts features using traditional 2D convolutional neural networks. The projection process results in the loss of point cloud geometry information. Therefore, the accuracy of semantic segmentation of the 3D point cloud is very poor. SEG-

Cloud divides the point cloud into a set of occupancy voxels. Then, 3D convolutional neural networks is applied to segment the point cloud scene. In this case, the voxel size is a very important hyperparameter, and the quantization error of voxelization will lead to the loss of geometric information. The high precision of voxel size will bring the burden of calculation. Point-based methods include ShellNet, GACNet, SPG, KPConv, and RandLA-Net, described in related work sections. Compared with other methods, RandLA-Net has better segmentation accuracy, and it is used as a representative to compare with our method. The results showed that LDE-Net was superior to all of them in terms of the mloU and the OA.

Figure 5 shows the semantic segmentation of LDE-Net and RandLA-Net in detail. The edges obtained by RandLA-Net are very rough and can cause some trouble. In labeling each point, LDE-Net adds a local feature aggregation classifier that takes into account the semantic information of the surrounding points. As a result, the edges obtained by LDE-Net are very smooth and precise. As shown in Figure 5, the red boxes are all details of how our method (LDE-Net) compares to RandLA-Net. Table 2 shows the accuracy of LDE-Net on the semantic3D dataset. Due to the addition of the global feature extraction modules and two local feature extraction modules, LDE-Net is 0.2% higher than RandLA-Net on the mIoU.

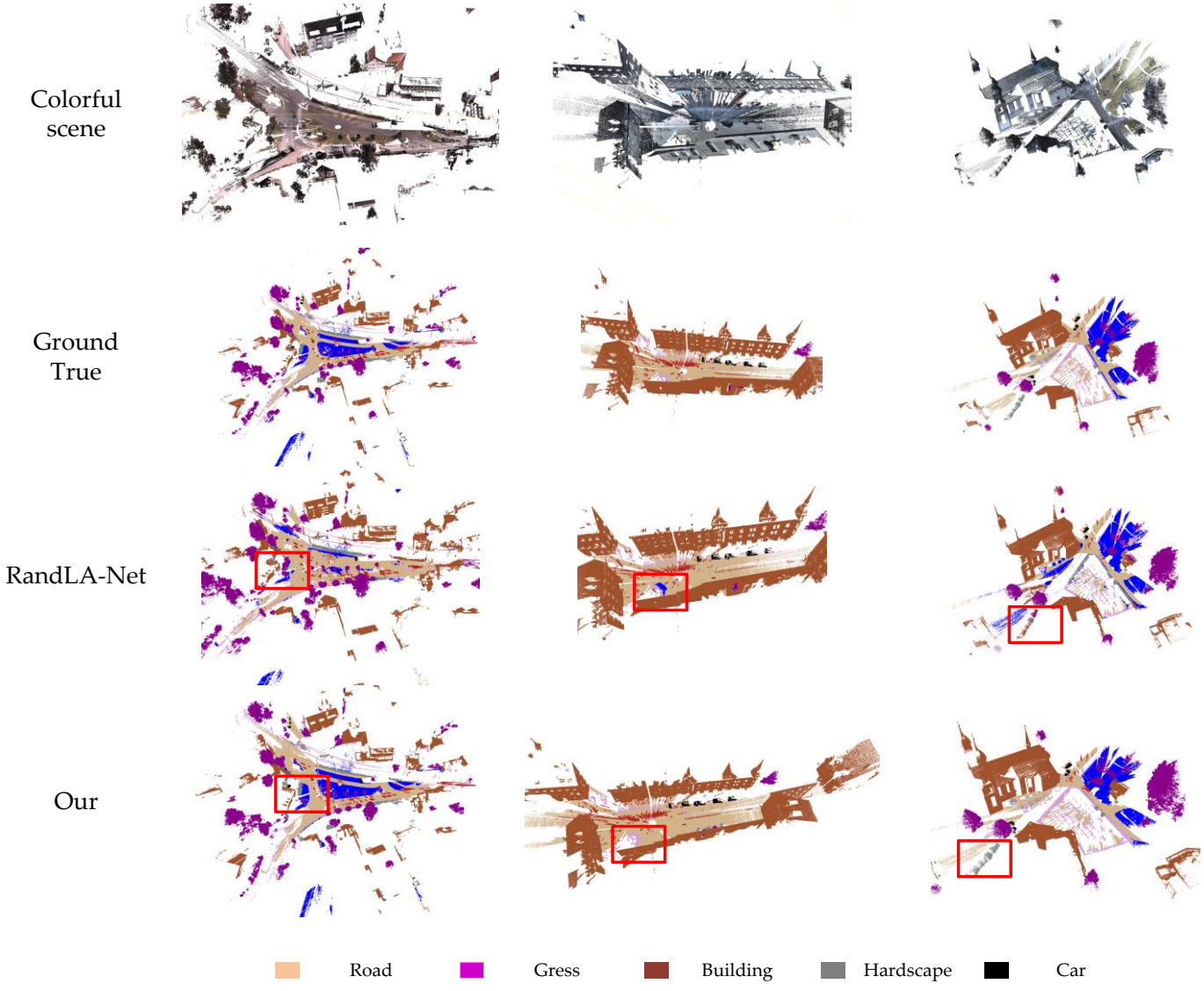

Figure 5. Semantic segmentation of colored 3D point clouds in semantic3D.

Table 2. Results of different methods in Semantic3D.

Methods	mIou	OA	Class Accuracy							
			Man-Made	Natural	HighVeg	Low Veg	Buildings	Hard Scape	Scanning	Cars
SnapNet_	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF_MSSF	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
ShellNet	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GAENet	70.8	91.9	86.4	77.7	88.5	60.6	94.2	37.3	43.5	77.8
SPG	73.2	94.0	97.4	92.6	87.9	44.0	84.2	31.0	63.5	76.2
KPConv	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	77.3	79.7
RandLA-Net	77.4	94.8	95.6	91.4	86.6	51.5	95.7	51.5	69.8	76.8
Our	77.6	95.0	96.8	90.5	85.1	52.3	97.5	54.3	71.2	79.5

4.4. SemanticKITTI

Table 3 reproduces the experimental results of several classic algorithms for semantic segmentation of 3D point clouds in SemanticKITTI dataset, including PointNet [24], PointNet++ [35], SPG [33], SqueezeSegV2 [34], RangeNet++ [19], and RandLA-Net [5].

Table 3. Results of different methods in SemanticKITTI.

Methods	mIou	Road	Sidewalk	Parking	Other-Ground	Building	Car	Truck	Bicycle	Motorcycle	Other-Vehicle	Vegetation	Trunk	Terrain	Person	Bicyclist	Motorcyclist	Fence	Pole	Traffic-Sign
PointNet	14.6	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPG	17.4	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
PointNet++	20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
SqueezeSegV2	39.7	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
RangeNet++	52.2	91.8	75.2	65.0	27.8	87.4	91.4	25.7	25.7	34.4	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9
RandLA-Net	53.9	90.7	73.7	60.3	20.4	86.9	94.2	40.1	26.0	25.8	38.9	81.4	61.3	66.8	49.2	48.2	7.2	56.3	49.2	47.7
Our	54.9	91.3	73.9	62.0	22.6	88.6	93.9	40.0	27.9	28.2	40.5	81.2	61.9	67.1	45.0	49.3	10.9	58.0	50.0	54.1

PointNet and PointNet++ are not suitable for 3D point cloud semantic segmentation in large scenes, so the accuracy is very poor. SqueezeSegV2 proposes a spatially adaptive convolution to better capture symbiotic relationships between objects in LiDAR images, and also uses KNN for post-processing. RandLA-Net is a 3D point cloud semantic segmentation for large scenes, which leads to its better effect, and it is used as a representative to compare with our method. RandLA-Net has a module for local feature extraction, so its semantic segmentation accuracy of some objects is relatively high. In contrast, our method has a global feature extraction module and two kinds of local feature extraction modules. Therefore, our method (LDE-Net) is higher than RandLA-Net in semantic segmentation accuracy for some objects and overall.

Figure 6 shows the results of semantic segmentation between our method (LDE-Net) and RandLA-Net on the SemanticKITTI dataset. It can be seen from the results that road and sidewalk are similar in some aspects. RandLA-Net does not distinguish well between road and sidewalk. Since LDE-Net has two kinds of local feature extraction modules and global feature extraction modules, it can better reference the neighborhood information so that these classes of semantic segmentation accuracy is higher. Table 3 shows the accuracy of LDE-Net on the semanticKITTI dataset. Due to the addition of the global feature extraction modules and two local feature extraction modules, LDE-Net is 1% higher than RandLA-Net on the mIoU.

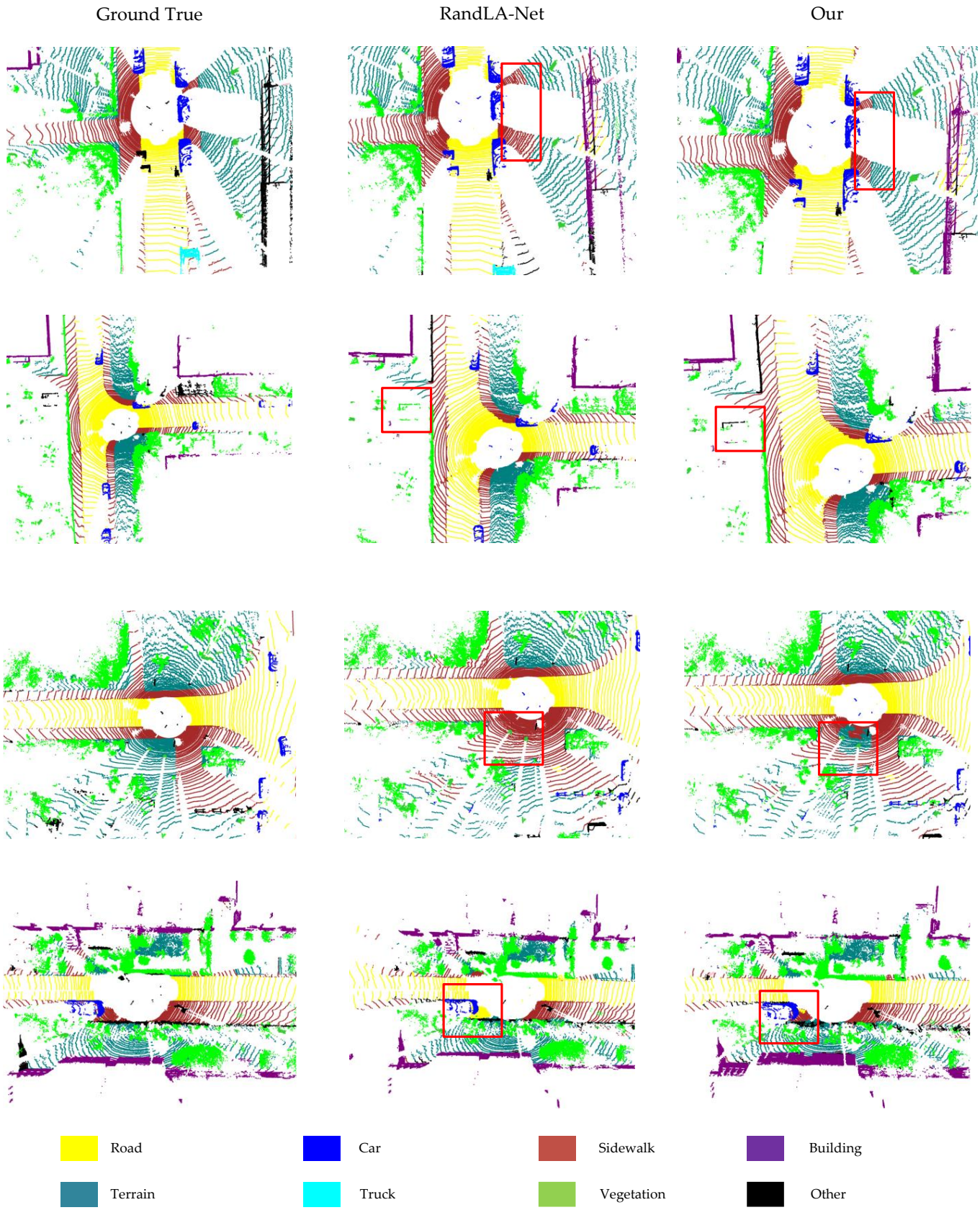

Figure 6. Semantic segmentation of 3D point clouds in semanticKITTI.

4.5. Complexity Comparison Experiment

We systematically evaluated the overall efficiency of PointNet, PointNet++, SPG, RandLA-Net, and LDE-Net on Sequence 08 of the SemanticKITTI dataset. Table 4 quantifies the total time, parameters, and mIoU of different methods. PointNet has the fewest parameters and faster inference times. Since PointNet is a feature extraction for the whole point cloud, it is not suitable for 3D point cloud semantic segmentation in large-scale environments. Then, PointNet++ added local feature extraction specifically, and the inference time, parameter number, and mIoU also increased a lot. SPG has fewer parameters, but it takes the longest time to process point clouds due to the expensive steps of geometric partitioning and supergraph construction. Compared with RandLA-Net, LDE-Net once again enhances the ability to extract local features. Its model complexity and inference time increased, but it has some advantages in terms of the mIoU. Thus, LDE-Net improves the mIoU by 1% with a slight increase in computation time.

Table 4. Experimental of model complexity and inference time.

	Total Time (Seconds)	Parameters (Millions)	mIoU (%)
PointNet	192	0.8	14.6
PointNet++	9831	0.97	20.1
SPG	43,584	0.25	17.4
RandLA-Net	185	1.24	53.9
LDE-Net (Our)	191	1.32	54.9

4.6. Ablation Experiments

The above comparative experiments demonstrate the advantages of the LDE-Net in the semantic segmentation of outdoor environments. In order to better understand the network and evaluate the role of each module, we conducted several ablation experiments as follows. We focused on two modules in LDE-Net: local-global feature extraction modules (LGFE) and the local feature aggregation classifier (LFAC). The purpose of ablation experiments is to investigate the impact of these two modules on the overall semantic segmentation accuracy of the network.

In Table 5, BaseNet represents the basic network without these two modules, LGFE represents the local-global feature extraction module, and LFAC represents the local feature aggregation classifier.

Table 5. Ablation experiment of LDE-Net on SemanticKITTI dataset.

	mIoU (%)	Total Time (Seconds)	Parameters (Millions)
Base-Net	50.1	173	1.10
Base-Net + LGFE	53.3	189	1.30
Base-Net + LFAC	52.1	190	1.27
Base-Net + LGFE + LFAC (LDE-Net)	54.9	191	1.32

As shown in Table 5, the mIoU of a network without LGFE is 2.8% lower than the LDE-Net with LGFE. The mIoU without LFAC module is 1.6% lower than that with LFAC module. Although the two modules will slightly increase the number of arguments and inference time, the increase in mIoU is significant. This shows that local feature aggregation classifiers and local-global feature extraction module can improve the context-aware ability of the classifier by referring to neighborhood features, thus improving the overall accuracy.

From the above experiments, it can be seen that each module in LDE-Net has its own meaning, and the combination can achieve the best effect. Both LGFE and LFAC can enhance the model's local perception ability. Therefore, it is meaningful to use all modules of LDE-Net for semantic segmentation in colored outdoor environment, and good results can be obtained.

5. Limitations and Future Work

The above experiments show that the mIoU accuracy of semantic segmentation and processing speed of our method have been improved to some extent, but the accuracy of small sample data is still very low. This leads to certain errors in visual understanding in driverless scenes, and has a certain impact on the judgment of driverless driving. In order to solve the forgetting problem caused by sample imbalance, some scholars have proposed the use of memory module to enhance the ability of small sample feature extraction. Our next task is to introduce this memory network into existing models to solve the problem of low accuracy for small samples.

For the same dataset of urban scenes, Semantic3D has location and color information, while SemanticKITTI only has location information. This results in a slight gap between SemanticKITTI's accuracy and Semantic3D's accuracy. Considering that many lIDARs in reality do not have the ability to collect color information. We consider calculating the neighborhood normal vector information of each point to enhance the information representation ability of the point cloud. The position information and normal vector information of the point cloud are sent to the network to enhance the accuracy.

6. Conclusions

Semantic segmentation of outdoor scene environments is a key technology to ensure the driving of unmanned vehicles. In this article, we propose a novel network LDE-Net for semantic segmentation of colored 3D point clouds in outdoor environments. The network is composed of local-global feature extraction modules and a local feature aggregation classifier. LDE-Net integrates a local-global feature extraction module into encoder to improve feature extraction capability. The local feature aggregation classifier is designed to classify the extracted features and consider the neighborhood information to enhance the local perception ability. The employment of these technologies will facilitate the deployment of unmanned vehicles in real world settings with greater efficiency. Experimental results demonstrate that the LDE-Net has better performance than other methods.

Author Contributions: Conceptualization, K.Z., Y.A. and Y.C.; Methodology, K.Z., Y.A. and Y.C.; Software, K.Z.; Validation, K.Z. and H.D.; Formal analysis, Y.A.; Investigation, Y.C.; Writing—original draft, K.Z. and H.D.; Writing—review & editing, K.Z., Y.A., Y.C. and H.D.; Visualization, K.Z.; Supervision, Y.A. and Y.C.; Project administration, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62173055 and 61673083, in part by the Natural Science Foundation of Liaoning Province under Grant 2023-MS-093, in part by the Science and Technology Major Project of Shanxi Province under Grant 20191101014, in part by the Major Science and Technology Project of Henan Province under Grant 231111222900, in part by the Joint Fund of Science and Technology Research and Development Plan of Henan Province under Grant 232103810038, and in part by the Key Research Projects of Higher Education Institutions of Henan Province under Grant 24A460009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets (Semantic3D and SemanticKITTI) were analyzed in this study. These data can be found here: [http://www.semantic3d.net (accessed on 19 December 2023), http://www.semantic-kitti.org (accessed on 19 December 2023)].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Koppula, H.; Anand, A.; Joachims, T.; Saxena, A. Semantic labeling of 3d point clouds for indoor scenes. In Proceedings of the Advances in Neural Information Processing Systems, Granada, Spain, 12–14 December 2011; pp. 244–252.
2. Tateno, K.; Tombari, F.; Navab, N. Real-time and scalable incremental segmentation on dense SLAM. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 4465–4472.
3. Deng, C.; Qiu, K.; Xiong, R.; Zhou, C. Comparative Study of Deep Learning Based Features in SLAM. In Proceedings of the 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Nagoya, Japan, 13–15 July 2019; pp. 250–254.
4. Li, J.; Li, Z.; Feng, Y.; Liu, Y.; Shi, G. Development of a Human–Robot Hybrid Intelligent System Based on Brain Teleoperation and Deep Learning SLAM. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1664–1674. [CrossRef]
5. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8338–8354. [CrossRef] [PubMed]
6. An, Y.; Li, B.; Hu, H.; Zhou, X. Building an Omnidirectional 3-D Color Laser Ranging System through a Novel Calibration Method. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8821–8831. [CrossRef]
7. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and Recognition Using Structure from Motion Point Clouds. In Proceedings of the Computer Vision—ECCV 2008, Berlin/Heidelberg, Germany, 12–18 October 2008; pp. 44–57.
8. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
9. Neuhold, G.; Ollmann, T.; Bulò, S.R.; Kontschieder, P. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5000–5009.
10. Che, Z.; Li, G.; Li, T.; Jiang, B.; Shi, X.; Zhang, X.; Lu, Y.; Wu, G.; Liu, Y.; Ye, J. D^2-City: A Large-Scale Dashcam Video Dataset of Diverse Traffic Scenarios. *arXiv* **2019**, arXiv:1904.01975.
11. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
12. Choi, Y.; Kim, N.; Hwang, S.; Park, K.; Yoon, J.S.; An, K.; Kweon, I.S. KAIST Multi-Spectral Day/Night Data Set for Autonomous and Assisted Driving. *IEEE Trans. Intel. Transp. Syst.* **2018**, *19*, 934–948. [CrossRef]
13. Chen, Y.; Wang, J.; Li, J.; Lu, C.; Luo, Z.; Xue, H.; Wang, C. LiDAR-Video Driving Dataset: Learning Driving Policies Effectively. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5870–5878.
14. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11618–11628.
15. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The ApolloScape Dataset for Autonomous Driving. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1067–10676.
16. Song, X.; Wang, P.; Zhou, D.; Zhu, R.; Guan, C.; Dai, Y.; Su, H.; Li, H.; Yang, R. ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5447–5457.
17. Yu, H.; Luo, Y.; Shu, M.; Huo, Y.; Yang, Z.; Shi, Y.; Guo, Z.; Li, H.; Hu, X.; Yuan, J.; et al. DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 21329–21338.
18. Boulch, A.; Saux, B.L.; Audebert, N. Unstructured point cloud semantic labeling using deep segmentation networks. In Proceedings of the Workshop on 3D Object Retrieval, Lyon, France, 23 April 2017; pp. 17–24.
19. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1887–1893.
20. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
21. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4213–4220.
22. Su, H.; Jampani, V.; Sun, D.; Maji, S.; Kalogerakis, E.; Yang, M.-H.; Kautz, J. SPLATNet: Sparse Lattice Networks for Point Cloud Processing. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2530–2539.
23. Rethage, D.; Wald, J.; Sturm, J.; Navab, N.; Tombari, F. Fully-Convolutional Point Networks for Large-Scale Point Clouds. In Proceedings of the Computer Vision—ECCV 2018, Cham, Switzerland, 8–14 September 2018; pp. 625–640.

24. Graham, B.; Engelcke, M.; Maaten, L.v.d. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
25. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.
26. Jiang, M.; Wu, Y.; Lu, C. PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation. *arXiv* **2018**, arXiv:1807.00652.
27. Engelmann, F.; Kontogianni, T.; Schult, J.; Leibe, B. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In Proceedings of the Computer Vision—ECCV 2018 Workshops, Cham, Switzerland, 8–14 September 2018; pp. 395–409.
28. Zhao, H.; Jiang, L.; Fu, C.W.; Jia, J. PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5560–5568.
29. Zhang, Z.; Hua, B.S.; Yeung, S.K. ShellNet: Efficient Point Cloud Convolutional Neural Networks Using Concentric Shells Statistics. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1607–1616.
30. Thomas, H.; Goulette, F.; Deschaud, J.E.; Marcotegui, B.; LeGall, Y. Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 390–398.
31. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.
32. Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; Shan, J. Graph Attention Convolution for Point Cloud Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10288–10297.
33. Landrieu, L.; Simonovsky, M. Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4558–4567.
34. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6410–6419.
35. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 5105–5114.