Large-Scale Semantic 3-D Reconstruction: Outcome of the 2019 IEEE GRSS Data Fusion Contest—Part B

Yanchao Lian, *Graduate Student Member, IEEE*, Tuo Feng, Jinliu Zhou,
Meixia Jia, *Graduate Student Member, IEEE*, Aijin Li, Zhaoyang Wu, Licheng Jiao, *Fellow, IEEE*,
Myron Brown, *Senior Member, IEEE*, Gregory Hager, *Fellow, IEEE*, Naoto Yokoya, *Member, IEEE*,
Ronny Hänsch, *Senior Member, IEEE*, and Bertrand Le Saux, *Member, IEEE*

***Abstract*—We present the scientific outcomes of the 2019 Data Fusion Contest organized by the Image Analysis and Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society. The contest included challenges with large-scale datasets for semantic 3-D reconstruction from satellite images and also semantic 3-D point cloud classification from airborne LiDAR. 3-D reconstruction results are discussed separately in Part-A. In this Part-B, we report the results of the two best-performing approaches for 3-D point cloud classification. Both are deep learning methods that improve upon the PointSIFT model with mechanisms to combine multiscale features and task-specific postprocessing to refine model outputs.**

***Index Terms*—Classification, convolutional neural networks, data fusion contest (DFC), deep learning, image analysis and data fusion, light detection and ranging (LiDAR), point cloud, semantic labeling, semantic mapping.**

I. Introduction

A CURRENT challenge of Earth observation is to add a new dimension to the representation of the world. Multiple 2-D imagery resources, with various sensors and resolutions, are available with which the surface of the Earth can be observed from above. However, for critical applications such as flight management and urban planning to environmental monitoring of forests, floods, and landslides, 3-D models of the ground are an essential source of insightful information.

Capturing 3-D information on large scales is challenging. Two categories of approaches are currently used: active and passive. Passive approaches include *structure from motion* and *multiple-view stereo* and leverage multiple satellite images corresponding to the same ground site to estimate common 3-D points. They yield in high-resolution and accurate elevation models and benefit from developments spanning more than four decades [1]. Active methods mostly refer to light detection and ranging (LiDAR) acquisitions. While satellite LiDAR is used for global low-resolution data collection, using airborne sensors during large aerial laser scanning (ALS) campaigns [2] is the preferred way to obtain detailed 3-D point clouds of the Earth's surface. Indeed, originating in the 1960s [3], ALS has a long history. Through continuous developments of data collection and processing methods [4], it has become the most accurate method to produce digital elevation models (DEMs) and 3-D models of the environment. However, automated understanding of these 3-D point clouds remains a challenge. This has resulted in numerous approaches for the classification of point clouds, object detection, and point cloud semantic segmentation (e.g., [5], [6] for recent, comprehensive reviews).

In 2019, the Johns Hopkins University (JHU) Applied Physics Laboratory (APL) joined with the Image Analysis and Data Fusion Technical Committee (IADF TC) of the IEEE Geoscience and Remote Sensing Society (GRSS) to organize a new benchmark on the topic of semantic 3-D. The IADF TC is an international network of scientists working on Earth observation, geo-spatial data fusion, and algorithms for image analysis. It aims at connecting people and resources, educating students and professionals, and promoting theoretical advances and best practices in image analysis and data fusion. Every year since 2006, it has organized a challenge for fostering ideas and progress in remote sensing, distributing novel data, and benchmarking analysis methods: the data fusion contest (DFC) [7]–[19].

The 2019 DFC (DFC19) aimed at *large-scale semantic 3-D reconstruction*, which encompasses both 3-D modeling of the Earth's surface from satellite imagery and automated mapping of its semantic aspects. Covering two sites and releasing a

Manuscript received July 31, 2020; revised October 22, 2020; accepted October 28, 2020. Date of publication November 2, 2020; date of current version January 6, 2021. The work of Yanchao Lian, Tuo Feng, Jinliu Zhou, Meixia Jia, Aijin Li, Zhaoyang Wu, and Licheng Jiao were supported in part by the State Key Program of National Natural Science of China under Grant 61836009, in part by the National Natural Science Foundation of China under Grant U1701267, and in part by the Major Research Plan of the National Natural Science Foundation of China under Grant 91438201. The work of Myron Brown was supported by IARPA under Contract 2017-17032700004. *(Corresponding author: Ronny Hänsch.)*

Yanchao Lian, Tuo Feng, Jinliu Zhou, Meixia Jia, Aijin Li, Zhaoyang Wu, and Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: yclian@stu.xidian.edu.cn; fengt@stu.xidian.edu.cn; zhoujinliu@stu.xidian.edu.cn; mxjia@stu.xidian.edu.cn; aijinli@stu.xidian.edu.cn; wuzhaoyang@stu.xidian.edu.cn; lchjiao@mail.xidian.edu.cn).

Myron Brown is with Johns Hopkins University Applied Physics Laboratory, Laurel, MD11100 USA (e-mail: myron.brown@jhuapl.edu).

Gregory Hager is with Johns Hopkins University, Baltimore, MD21218 USA (e-mail: hager@cs.jhu.edu).

Naoto Yokoya is with the Graduate School of Frontier Sciences, University of Tokyo, Chiba 277-8561, Japan, and also with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: naoto.yokoya@riken.jp).

Ronny Hänsch is with German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: rww.haensch@gmail.com).

Bertrand Le Saux is with ESA/ESRIN, ϕ-Lab,, (RM) I-00044 Frascati, Italy (e-mail: bls@ieee.org).

Digital Object Identifier 10.1109/JSTARS.2020.3035274

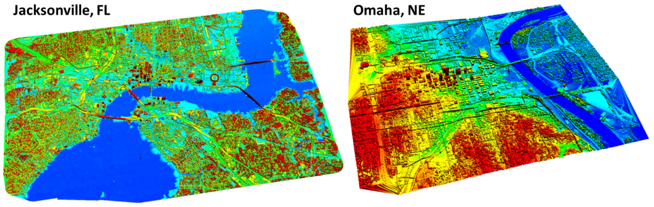


Fig. 1. Airborne LiDAR for DFC19 Track 4 challenge.

large volume of airborne LiDAR data with semantic labels, the competition was highly ambitious and addressed a current scientific issue: the semantic labeling of 3-D ALS point clouds.

DFC19 made use of the Urban Semantic 3-D (US3D) data [20] to deliver an unprecedented amount of more than 320 GB of images and 3-D reference data spanning over roughly 20 km^2 of the urban areas of Jacksonville (Florida) and Omaha (Nebraska) in the United States. In detail, it comprised of the following.

1) WorldView-3 satellite images (courtesy of Maxar), both panchromatic and eight-band visible and near-infrared, with ground-sampling distances of 35 cm and 1.3 m, respectively.
2) 3-D data provided as point clouds or digital surface models (DSMs) produced from airborne LiDAR, with a resolution of 80 cm.
3) Semantic labels for urban classes: buildings, elevated roads and bridges, high vegetation, ground, and water.

The DFC19 was organized as four parallel tracks: Tracks 1, 2, and 3 were dedicated to semantic 3-D reconstruction with various levels of input data. Participants were able to submit semantic maps and DEMs resulting from single-view semantic 3-D methods (Track 1), two-view stereo semantic 3-D methods (Track 2), and multiview stereo semantic 3-D algorithms (Track 3). Track 4 addressed a related but different problem: large-scale 3-D point cloud semantic labeling.

The present article is the second of a two-part manuscript that aims at presenting and critically discussing the scientific outcomes of the 2019 contest. Part A [21] focuses on semantic 3-D reconstruction and covers Tracks 1, 2, and 3. Part B is dedicated to large-scale point cloud classification and reports on Track 4.

We describe the relevant parts of the dataset in Section II and discuss the overall results of the 3-D point cloud segmentation challenge of the contest in Section III. Then, we will focus in more detail on the approaches proposed by the winning teams in Sections IV and V. Finally, Section VI concludes this article.

II. Data of the Point Cloud Classification Challenge of the DFC 2019

For all DFC19 challenge tracks, we provided data from US3D, a large-scale public dataset including multidate, multiview, and multiband satellite images and reference geometric and semantic labels covering approximately 100 km^2 over Jacksonville, Florida and Omaha, Nebraska, United States [20] (see Fig. 1). For the contest, we provided training and test datasets for each challenge track, including approximately 20% of the US3D data.

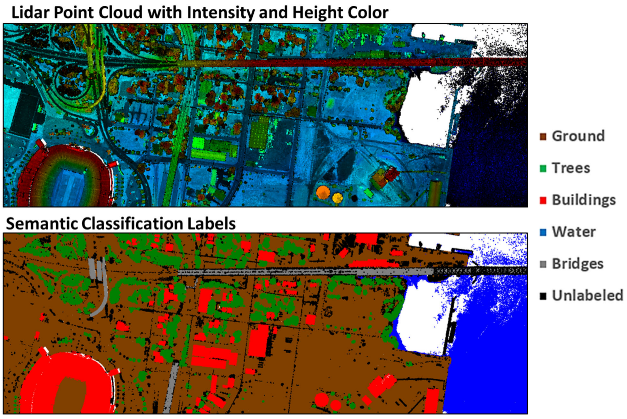

Lidar Point Cloud with Intensity and Height Color
Semantic Classification Labels
Ground
Trees
Buildings
Water
Bridges
Unlabeled

Fig. 2. Example LiDAR point cloud and reference semantic labels.

Details of the data provided for semantic 3-D reconstruction (Tracks 1, 2, and 3) are presented in Part A [21]. For Track 4, we provided airborne LiDAR point clouds with approximately 80 cm aggregate nominal pulse spacing. Point clouds were provided in text files with the format {x, y, z, intensity, return number}, and semantic labels were provided in text files with format {label}, similar to the formats used for the Semantic 3-D dataset [22]. Semantic classes in the contest (with label number indicated from the LAS specification) included ground (2), trees or high vegetation (5), buildings (6), water (9), and elevated roads or bridges (17), as shown in Fig. 2. We provided semantic labels for the training regions only. For the validation and test regions, only LiDAR point clouds were provided. The ground truth for the validation and test sets remained undisclosed and were used for evaluation of the results.

The training and test sets for Track 4 include LiDAR point clouds for each geographic 500 × 500 m tile (111 tiles for the training set, 10 tiles for the validation set, and 10 tiles for the test set). Training and test datasets were selected to ensure similar semantic and geometric distributions, as shown in Fig. 3. After completion of the contest, we also released an extended training dataset including semantically labeled LiDAR point clouds for 806 geographic tiles. This contest data and extended training data are available on IEEE DataPort [23], [24].

With respect to other reference point cloud datasets and benchmarks of the community, DFC19 offers the ability to provide new insights on point cloud classification. It is 80 times larger than the ISPRS 3-D Semantic Labeling Contest (Vaihingen3D) [25] and addresses the specificities of ALS. This is in contrast to established static or mobile laser scanning datasets, such as IQmulus [26], Paris-Lille-3D [27] or Semantic 3-D [22], which are captured from the ground.

III. Organization, Submissions, and Results

Section III-A describes Track 4 of DFC19, which was dedicated to 3-D point-cloud classification. We then analyze the participation in Section III-B and the winning approaches in Section III-C.

TABLE I
TOP RANKED TEAMS AND APPROACHES

Rank	Team	mIoU	Affiliation	Approach		
				Data augmentation	Ensemble	Postprocess.
1	fengzicai	0.9455	Xidian University	✓	✓	✓
2	aijinli0613	0.9454	Xidian University	✓		✓
3	Shouping	0.9443	Xidian University			✓
4	liwuzhao	0.9302	Wuhan University		✓	✓
–	Baseline	0.8644	–			

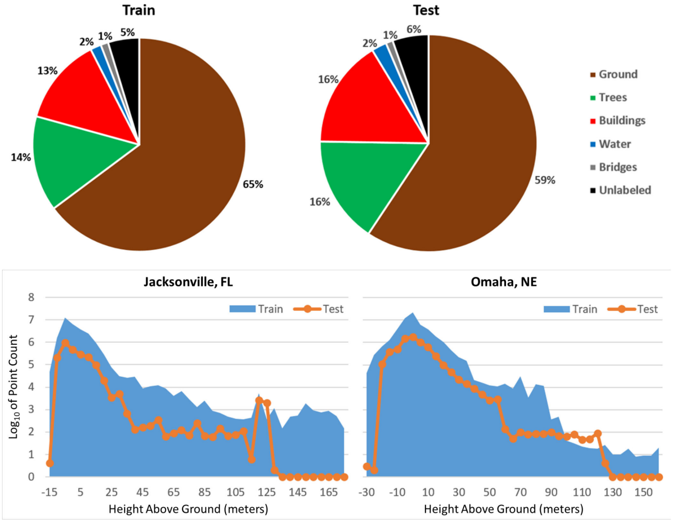


Fig. 3. Distribution of semantic labels for train and test sets and height above ground values for Jacksonville and Omaha.

A. Track 4: 3-D Point Cloud Classification

For each geographic tile, LiDAR point cloud data were provided. The objective was to predict a semantic label for each 3-D point. Participants of Track 4 submitted 3-D semantic predictions in text files similar to the text files of the training set. Performance was assessed using mean Intersection over Union (mIoU), while overall accuracy (OA) metrics were reported as well. We developed and evaluated a baseline algorithm using the popular PointNet++ deep network [28] and made it available on GitHub for the contestants [29].

B. Participation

There were 710 unique registrations for downloading the DFC19 data from 45 countries. In total, 47 teams participated in Track 4 on the Codalab competition website during the test phase. This number is larger than those of the other tracks, indicating enthusiasm in the remote sensing community to tackle large-scale 3-D point cloud classification tasks.

C. Best-Performing Approaches and Discussion

The first- and second- ranked teams were awarded as winners and presented their solutions during the 2019 IEEE International Geoscience and Remote Sensing Symposium in Yokohama, Japan. The winning teams were:

1) *1st place*: *fengzicai* team; Yanchao Lian, Tuo Feng, and Jinliu Zhou from Xidian University, China; Dense PointNet++ architecture with PointSIFT module (DPNet) [30].
2) *2nd place*: *aijinli0613* team; Meixia Jia, Zhaoyang Wu, and Aijin Li from Xidian University, China; Global PointSIFT Attention Network (PointSIFT-GPA) [31].

Table I summarizes the four top-ranked teams and their approaches. All of the top-ranked teams extended or modified well-established architectures (e.g., PointNet++ and PointSIFT) for semantic point labeling by embedding skip connections or attention modules. The winners performed data augmentation to deal with the class imbalance problem. Similar to the other tracks and to previous editions of the DFC, ensembling (or model fusion) and postprocessing played an important role for further performance improvement.

In Sections IV and V, we present the solutions proposed by the two top-ranked teams of Track 4, respectively. We detail the winning classification methodologies and provide an in-depth analysis of the advantages and limitations of the solutions.

IV. FIRST PLACE IN THE POINT CLOUD SEMANTIC LABELING CHALLENGE: LIAN–FENG–ZHOU TEAM, XIDIAN UNIVERSITY

Here, we describe the winning algorithm from the point cloud semantic labeling challenge. We developed a Dense PointNet++ architecture [28] with PointSIFT modules [32], called DPNet, for semantic segmentation of 3-D point cloud data. Specifically, we enhanced the features of each layer to obtain more abundant point cloud features using multiple nested sampling layers. Short and long skip link concatenations were introduced in the network to bridge the semantic gap. The cross-entropy loss function with variable weights is utilized during training to reduce the problem of data imbalance. Averaging the multiple outputs provided by DPNet as well as applying a grid map based correction of the estimates are used to obtain a more reliable performance. Experimental results show that DPNet is superior to state-of-the-art network architectures in 3D point cloud semantic segmentation.

A. Method: DPNet and Grid Map

1) DPNet. Network Architecture: Our method makes use of an encoder–decoder network structure that is most commonly used for 3-D semantic segmentation, e.g., in state-of-the-art networks such as PointNet++ [28] and PointConv [33]. To establish the transfer of feature information, most of these networks directly combine the shallow features in the encoder with the deep features in the corresponding decoder. However, the dissimilarity of the semantic information of the combined

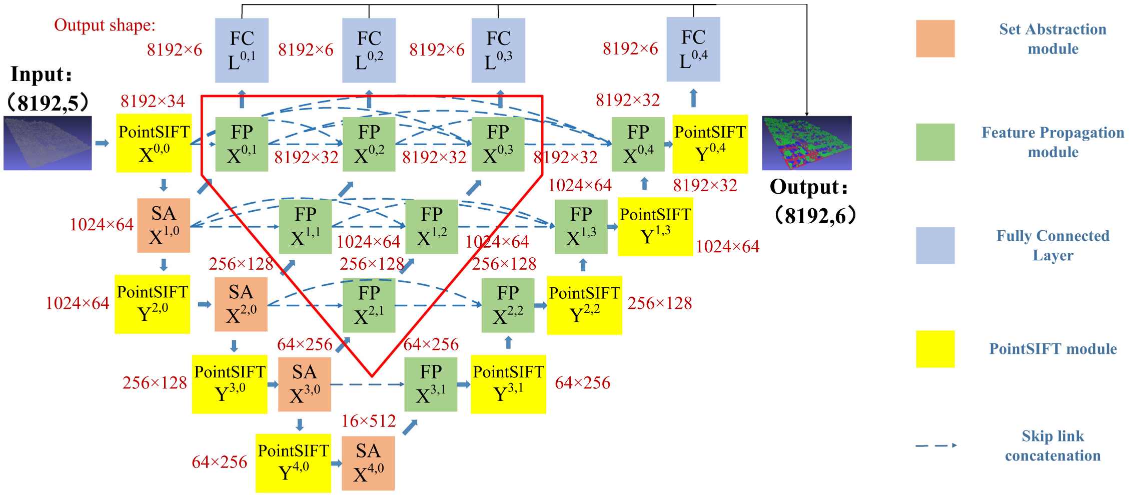


Fig. 4. Model structure of DPNet, which has a dense PointNet++ architecture. The structure includes multiple nested up-sampling layers and a series of short and long skip link concatenation, which makes a mutual connection between the down-sampling layer and all corresponding up-sampling layers. Red modules indicate SA modules, green modules indicate FP modules, blue modules indicate the fully connected layers, yellow modules indicate PointSIFT modules, and the dotted lines indicate the short and long skip link concatenation.

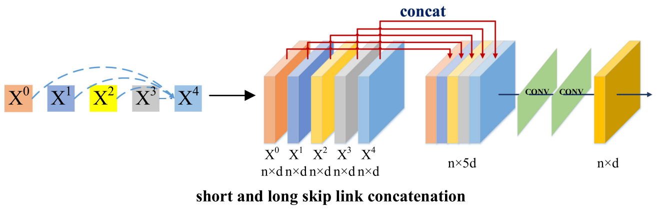


Fig. 5. Illustration of short and long skip link concatenation. Features with different semantic information are connected by a series of short and long skip link concatenation, and then through a set of convolutions. The complete and rich semantic features are obtained.

features results in a semantic gap. To bridge this gap, we developed DPNet, a Dense PointNet++ architecture for semantic segmentation of 3-D point cloud data. Fig. 4 shows the network structure of DPNet, with major changes in the following three aspects:

1) a feature propagation (FP) module is inserted behind each set abstraction (SA) module;
2) SA modules and corresponding FP modules of all levels are connected through a series of dense, nested, complete, short and long skip link concatenation;
3) the network has multiple outputs, each providing supervision at training time.

Because shallow and deep features are equally important, both should be upsampled. Shallow semantic features catch simple information of a point cloud such as spatial structure and the relationship between neighboring points. Deep semantic features exploit more semantic information due to larger receptive fields and a larger number of convolution layers. More comprehensive semantic features can therefore be obtained by using both, shallow and deep features. To this aim, an FP module is inserted behind each SA module (red box in Fig. 4). At the same time, the SA module and corresponding FP modules of all levels are connected through short and long skip link concatenation (see Fig. 5), which helps to accumulate and integrate the previous features into the latter features. High and low level semantic features of the same size are concatenated by several shortcuts and then transformed by a set of convolutions. This yields a d dimensional output of more similar semantic features with a reduced semantic gap. DPNet uses PointSIFT and SA modules with multi-scale grouping (MSG) [28]. (1) reflects the relationship between each module

$$x^{i,j} = \begin{cases} D(x^{i-1,j}), i > 0, j = 0 \\ \text{mlp}\left(\left[\left[x^{i,n}\right]_{n=0}^{j-1}, U\left(x^{i+1,j-1}\right)\right]\right), j > 0 \end{cases} \qquad (1)$$

where $x^{i,j}$ denotes the feature output of module $X^{i,j}$; i and j denote the index of SA module passed in the encoder and the index of FP module in the corresponding decoder, respectively; $D(\cdot)$ denotes a down-sampling layer comprised of PointSIFT and SA modules; mlp denotes a multilayer perceptron (MLP), $U(\cdot)$ is an up-sampling layer; $[\cdot]$ denotes short and long skip link concatenation; and $x^{0,0}$ are the features extracted by the PointSIFT module. In (1), when $i > 0$ and $j = 0$, $x^{i,j}$ is only obtained from the output of the previous down-sampling layer. When $j > 0$, calculating $x^{i,j}$ requires not only an output, which is obtained by $x^{i+1,j-1}$ passing through an up-sampling layer, but also j outputs, which have the same index i of $x^{i,j}$ and have short and long skip link concatenation with the output from the previous up-sampling layer.

During both, the training and inference stage, the point clouds in each scene are partitioned into several blocks, and a fixed number (8192) of points are sampled from each block as one batch input. The input of the network is a 5-D vector as described in Section II. In the encoder, the input features (8192, 5) pass through four successive down-sampling layers (PointSIFT + SA module) to obtain features with sizes of (1024, 64), (256, 128), (64, 256), and (16, 512), respectively. These features have large receptive fields. In the decoder, the output of each SA module needs to be decoded by a series of consecutive FP modules, which include short and long skip link concatenations.

These concatenations connect the down-sampling layer and all corresponding up-sampling layers. Finally, a fully connected layer and softmax classifier are used to predict the semantic label.

$L^{0,1}$, $L^{0,2}$, $L^{0,3}$, and $L^{0,4}$ denote the four outputs of DPNet, each of which is a c-dimensional vector $\{p_1, p_2, \ldots, p_k, \ldots, p_c\}$, where p_k is the predictive probability of class k. Each of these outputs is the result of a semantic segmentation task. We use the average of the four outputs as the final output of the network to obtain a higher accuracy. Similarly, the total loss function of the network is the average of the cross-entropy loss of each output with a variable weight defined as

$$L = -\frac{1}{B}\sum_{b=1}^{B} W_b \cdot \log Y_b \tag{2}$$

where Y_b denotes the prediction probability value of sample b after passing the softmax classifier; the batch size is represented by B; W_b is the corresponding weight of sample b and is computed as $W_b = 1/ln(1.2 + n/N)$, where n denotes the number of points with the same category as sample b and N denotes the number of all point clouds. While W_b is invariant in the original weighted cross-entropy function, we use weights that vary with the deviation between the predicted probability value and the reference data. W_b should take the maximum weight between the reference data and the misjudged label when the predicted value is not equal to the reference data. The activation function used is the ELU function [34] instead of the ReLU function.

Data Preprocessing. Data preprocessing includes normalization and data augmentation. The airborne 3-D point cloud data is standardized according to the average and variance values of the feature values of all point clouds in the ith 3-D point cloud data scene. The eigenvalue r_i^j of the jth dimension in the ith 3-D point cloud scene is normalized through

$$r_i^{j'} = \frac{r_i^j - \mu_i^j}{\sqrt{\left(\sum_{i=0}^{k} \delta_i^j \cdot N_i\right) / \sum_{i=0}^{k} N_i}} \tag{3}$$

where N_i is the number of point clouds in the ith 3-D point cloud data scene, k is the total number of scenes, $r_i^{j'}$ denotes the normalized eigenvalue, μ_i^j and δ_i^j denote the average and variance values of the jth dimension feature values of all point clouds in the ith 3-D point cloud data scene, respectively. Eigenvalue normalization can effectively eliminate the problems of inconsistent point cloud coordinate systems and deviations in the same category of eigenvalues in different scenes due to changes in sensing height and area and can greatly improve the accuracy for model training. Redundant points in the normalized 3-D point cloud data are evenly deleted to reduce the computational load during training.

It is essential to improve the robustness of the model by data augmentation when the data samples are unbalanced. The imbalance of samples is a common problem in large-scale urban 3-D point cloud datasets. For example, the majority of points are labeled as ground, while the number of points labeled as water or elevated road are very few. The steps of data augmentation are shown in Fig. 6. First, all points in the 3-D point cloud data scene are divided into N blocks. The number of points in each block is no more than 65 536. If the number of ground points in a block exceeds 80% the entire block is deleted. If the proportion of points belonging to elevated road or water in the block is more than 10%, a reversed and rotated version of the block is added to the dataset.

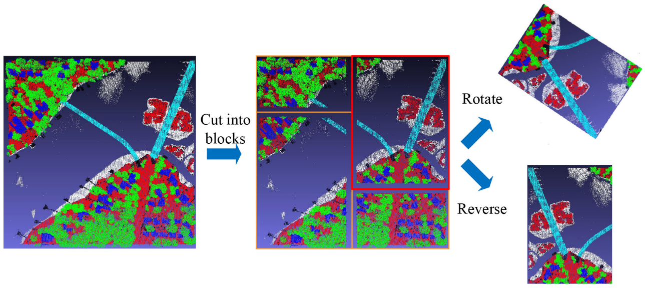


Fig. 6. Illustration of data augmentation. Points in the 3-D point cloud data scene are cut into blocks, and the number of categories in each block is counted. Then blocks with too large a proportion of points belonging to ground are deleted, and blocks with many samples of minority classes (elevated road and water) are randomly rotated and mirrored.

2) DPNet With Grid Map and Model Fusion: To further improve the performance of 3-D point cloud semantic segmentation, the results of DPNet are further processed by applying a grid map based correction as well as model fusion.

Grid Map. The usage of a grid map aims at correcting the segmentation result of the network. Intensity maps often clearly reflect the geomorphological features of the scene. We use intensity information as a prior to guide the correction of misclassification in the prediction results of DPNet, such as buildings being classified as ground or bridge. Fig. 7(g) shows an example where most of the misclassified points are corrected and the segmentation performance is improved significantly. The general grid map method is as follows.

1) Based on the prediction results of DPNet, the points belonging to trees are deleted since trees will affect the gradient around the boundary of buildings and the ground.
2) The intensity map, the gradient map, and the label map are obtained according to the intensity, gradient, and predicted label of each point cloud. The maximum intensity of points (with outliers excluded) in a grid cell is normalized and used as the pixel value to generate an intensity map, as shown in Fig. 7(b). The label with the largest number of points in each grid cell is taken as the label of the cell and the labels of all grid cells generate the label map. The gradient is the mean of the z value difference between a point and the surrounding k points in the x, y dimensions. The maximum gradient of all points in each cell is taken as the gradient value of the cell. Finally, the calculation of the binary gradient map can be described as follows:

$$G[i,j] = \begin{cases} 1, \text{ if } MAX\left[\frac{\sum_{k=0}^{K}(z_k - z_m)}{K}\right]_{\mathrm{m}} > 0.25 \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

where K is the number of points selected in x, y dimensions, and M denotes the number of points in $[i, j]$.

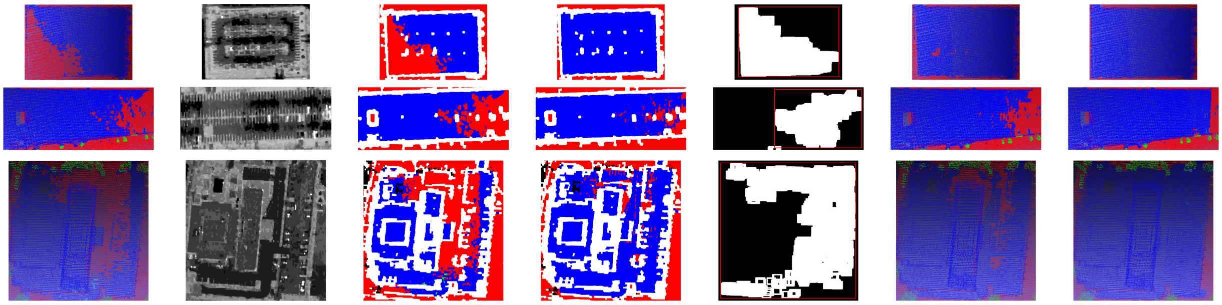


Fig. 7. Correction of misclassifications with grid maps [30]. (a) Segmentation results of DPNet. (b) Intensity map. (c) Mixed map. (d) Modified map. (e) Modified area. (f) Modified misclassified points. (g) Final modified result.

3) The gradient map is combined with the label map to generate a mixed map as shown in Fig. 7(c). To generate the mixed map, pixels with a value of 1 in the gradient map remain unchanged, while pixels with a value of 0 are replaced with that of the label map.
4) The label statistics for regions enclosed by white closed borders are computed. When the number of building pixels exceeds 40% of the total number, all pixels at the border will be modified as buildings. A 3×3 neighborhood is used to search for the corresponding pixels on the intensity map for all building pixels on the mixed map. The modified map is shown in Fig. 7(d), where blue pixels represent the building label.
5) The pixels of the white border on the modified map are replaced with those on the label map.
6) The modified map is compared to the mixed map. Regions with large changes are used as modified area [see Fig. 7(e)]. The grid map is mapped to 3-D space and the misclassified point cloud labels in the modified area are modified as shown in Fig. 7(f).
7) We use k-nearest-neighbors (kNNs) to further correct the misclassification as shown in Fig. 7(g). If the z value of a point in the modified area is larger than the average z value of all the ground label points, but it is found near building points, its label is changed to building.

Model Fusion. Based on the data augmentation for different labels during training, we obtained different models that are sensitive to the corresponding labels. The labels predicted by a model trained on the original data are fixed by the corresponding labels from models that are sensitive to trees and water, respectively. In addition, for the outputs of two models with different sensitivities to each label, confidence voting based on the softmax value is used. In this way, model fusion aggregates the output of multiple models to improve the performance of 3-D point cloud semantic segmentation.

B. Results and Discussion

1) Experimental Setup: Dataset. We performed statistical analysis on the sample categories in the training set of the US3D dataset. Points in the "ground" category occupy more than 60% of the entire training set, while points in the water or elevated road categories account for only 2% and 1%, respectively (i.e., data samples are extremely unbalanced).

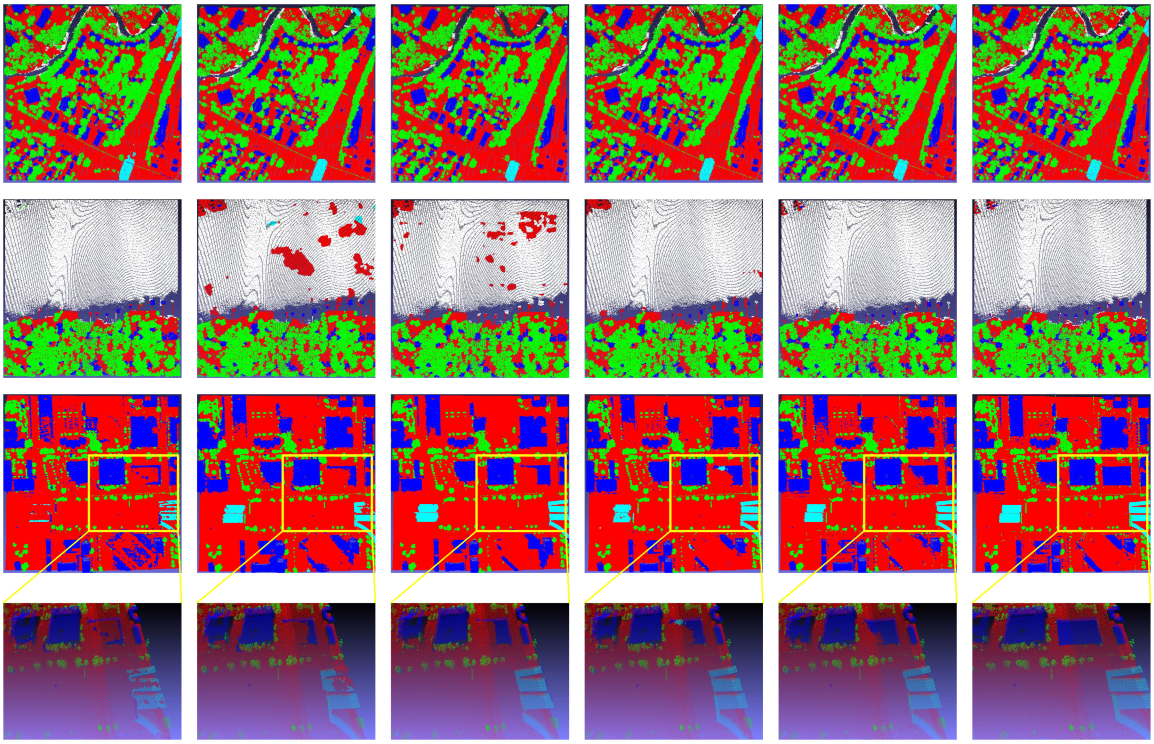
TABLE II
QUANTITATIVE COMPARISONS BASED ON PER-CATEGORY IOU, OA, AND MIOU OF DIFFERENT METHODS ON THE US3D DATASET BY SEMANTIC CATEGORY [WITH LABEL NUMBER INDICATED FROM THE LAS SPECIFICATION: GROUND (2), TREES OR HIGH VEGETATION (5), BUILDINGS (6), WATER (9), AND ELEVATED ROADS OR BRIDGES (17)]

Method	2	5	6	9	17	OA	MIoU
Test on US3D validation dataset for different methods							
PointNet++	0.965	0.938	0.885	0.72	0.697	0.967	0.841
PointSIFT	0.975	0.944	0.908	0.866	0.785	0.976	0.896
DPNet	0.980	0.942	0.914	0.886	0.835	0.978	0.912
Test on US3D test dataset for different methods							
DGCNN	0.964	**0.962**	0.863	**0.962**	0.487	0.969	0.848
PointNet++	0.954	0.952	0.837	0.884	0.766	0.963	0.879
PointNet++ (MSG)	0.968	0.949	0.858	0.931	0.748	0.969	0.891
PointCNN	0.967	0.954	0.883	0.883	0.835	0.973	0.904
PointSIFT	0.974	0.961	0.884	0.915	0.793	0.975	0.906
PointCONV	0.976	0.955	0.891	0.921	0.763	0.976	0.901
DPNet	0.979	0.958	0.900	0.949	0.865	0.979	0.93
Test on US3D test dataset for different DPNet architectures							
Fig. 9(b)	0.972	0.956	0.870	0.935	0.755	0.974	0.898
Fig. 9(c)	0.977	0.960	0.898	0.908	0.851	0.978	0.919
Fig. 9(d)	0.977	0.958	0.894	0.944	0.866	0.978	0.928
Test on US3D test dataset for different post processing							
DPNet+grid map	0.985	0.960	0.924	0.953	0.869	0.984	0.938
DPNet+grid map+model fusion	**0.987**	0.961	**0.933,**	0.955	**0.890**	**0.986**	**0.945**

Training Details. For a fair comparison, we trained each network model on TensorFlow with a single GeForce GTX TITAN X. The training batch size was set to 8 and the number of input points was 8192. We utilized the adaptive moment estimation (Adam) optimizer [35] with an initial learning rate of 0.001 to train all models. The number of training iterations was 66,300.

2) Qualitative and Quantitative Comparisons: Different networks were trained on the US3D dataset. Ten scenes in the training set were selected as the local verification set. The DPNet training process was smoother than that of other networks and resulted in the best performance with respect to OA and mIoU (as shown in the first part of Table II), although its convergence rate was not as fast as that of PointNet++.

The second part of Table II, as well as Fig. 8, show a comparative analysis of different methods. While all methods were



(a) DGCNN (b) PointNet++ (c) PointCNN (d) PointSIFT (e) DPNet (f) All methods

Fig. 8. Visualization of semantic segmentation results of each method applied to the US3D test set. (a) Segmentation results of DGCNN. (b) Segmentation results of PointNet++. (c) Segmentation results of PointCNN. (d) Segmentation results of PointSIFT. (e) Segmentation results of DPNet. (f) Segmentation results of our methods (DPNet with grid map and model fusion). Red points, green points, blue points, white points and cyan points denote ground, vegetation, building, water, and elevated road, respectively. The area in the yellow frame is enlarged to better observe the segmentation results.

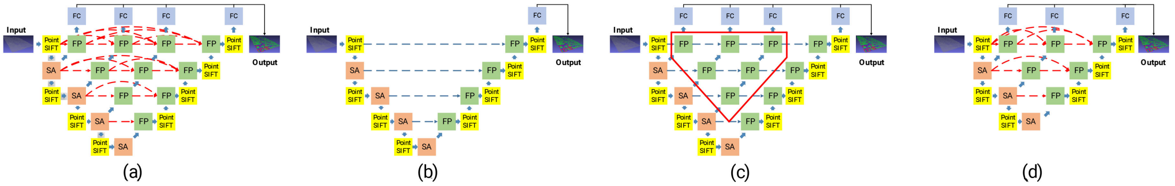


Fig. 9. Comparison of different network structures. (a) Model structure of DPNet. (b) Model structure of PointNet++ with the PointSIFT module. (c) Model structure of DPNet without the long skip link concatenation. (d) Model structure of DPNet with fewer modules.

able to segment airborne point cloud data in large scenes, their performance for each category was different. While Fig. 8(a) has the best segmentation result for "water" among all six methods, results for "bridge" are considerably weaker. This is because Fig. 8(a) models the relationship between points with their spatial difference. However, this difference only reflects the distance between the points and ignores the direction information. Fig. 8(c) uses the X-CONV operator to extract local features and has the best segmentation result for "bridge." Comparing Fig. 8(e) with Fig. 8(a), Fig. 8(b), Fig. 8(c), and Fig. 8(d), Fig. 8(e) achieve the best segmentation result in both the OA and mIoU on the US3D dataset. The introduction of grid maps in Fig. 8(f) improves the segmentation result of both "building" and "bridge."

3) Network Structure Analysis: We analyze the point cloud segmentation performance for four network structures. Fig. 9(a) is a complete DPNet structure with four output branches. Fig. 9(b) is PointNet++ structure with PointSIFT modules and only has one output branch. Fig. 9(c) adds three output branches on the basis of Fig. 9(b) and uses short concatenations to concatenate different branches. Fig. 9(d) shows a simple DPNet structure with three output branches. The third part of Table II shows the quantitative comparisons between these four network structures. The comparison of Fig. 9(b) and (c) shows that shallow features and deep features are equally important. The added up-sampling branches make effective use of shallow features, which are generated by down-sampling. The comparison of Fig. 9(a), (c), and (d) shows the impact of long and short concatenations. Three branches in Fig. 9(d) achieve better performance than four branches in Fig. 9(c), indicating that our network does not rely on the stacking of network parameters to improve performance. The improvement of performance is

TABLE III
MODEL SIZE AND FORWARD TIME OF DEEP ARCHITECTURES FOR 3-D POINT CLOUD SEMANTIC SEGMENTATION

	Model Size(MB)	Forward Time(ms)
PointNet++ [28]	3.69	423
PointNet++(MSG)	10.31	430
DGCNN [37]	9.07	1964
PointCNN [36]	43.91	770
PointConv [33]	82.62	9413
PointSIFT [32]	51.6	562
DPNet	32.23	515

due to the feature reuse achieved by the introduction of long and short concatenations. The performance of the four branches network in Fig. 9(a) is better than that of the three branches network in Fig. 9(c), also showing that the use of long and short concatenations improve performance.

4) Ablation Study: The experimental results in the fourth part of Table II show that better prediction results can be obtained by using a grid map to correct the segmentation results of DPNet and fusing models with different sensitivities for each category.

Since the 3-D point cloud data of the US3D dataset was obtained from aerial photography, the shape information of the objects is limited. From a bird's eye view, a flat roof is very similar to flat ground. Moreover, the height fluctuation between the ground is very large, i.e., the height of the ground in some scenes is higher than the height of the buildings in other scenes. Therefore, using neural networks to learn 3-D point cloud features will often misclassify the flat top of a large area as the ground.

The proposed grid map method successfully addressed these problems as shown in Table II. While the mIoU of vegetation, water, and elevated road did not change significantly, the mIoU of the building class increased from 0.90 to 0.92.

Fusing models with different prediction accuracy for points belonging to different classes further improves the overall accuracy.

The results in Fig. 8(e) and (f) illustrate the effect of grid map and model fusion. For example, in the third row, areas surrounded by the yellow square include some points that belong to buildings but can easily be identified as ground. With correction using the grid map, these points are predicted almost completely correctly.

5) Time and Space Complexity Analysis: Table III summarizes the model size and the forward time of DPNet and other related neural network algorithms. We record forward time with a batch size 8 using TensorFlow 1.12.0 with a single GeForce GTX TITAN X.

It is worth noting that compared with PointSIFT [32], DPNet (due to a reduced number of parameters in each layer) has fewer parameters and is faster. Models such as DGCNN [36] need to calculate the distance between each sampling point and its neighbors. Although the model size is small, it needs a large forward time. In contrast, DPNet achieves state-of-the-art performance on the US3D dataset and is efficient in terms of both time and memory.

V. SECOND PLACE IN THE POINT CLOUD SEMANTIC LABELING CHALLENGE: JIA–LI–WU TEAM, XIDIAN UNIVERSITY

In this section, we describe our algorithm that ranked second for the point cloud semantic labeling challenge. Our network can be viewed as an extension of PointSIFT [32] with global point attention (GPA). We first introduce the GPA module and then explain the network architecture and training of PointSIFT-GPA. Finally, we describe the result revise algorithm to postprocess our segmentation.

A. Method: PointSIFT-GPA

Network Architecture. In 2-D semantic segmentation, attention mechanisms have become an integral part of models that capture context information [38],[39]. Particularly, in global attention mechanisms [38], high-level features with rich semantic information help low-level features to refine resolution details. We applied global attention on 3-D point cloud data to fuse hierarchical features in a top–down manner and achieve an end-to-end trainable framework, denoted as PointSIFT-GPA network. The architecture of PointSIFT-GPA is illustrated in Fig. 10.

GPA. The GPA module has two inputs: low-level features $L_{n,d}$ and high-level features $H_{n,d}$ from the encoder and decoder, respectively. An 1×1 convolution is first applied to each low-level feature to increase the number of channels. The global high-level vector is generated from high-level features through a global average pooling function AVG$(H_{n,d})$. We generate the weighted global high-level vector $W_{n,1}$ via softmax, i.e.,

$$W_{n,1} = \frac{\exp[\text{AVG}\,(H_{n,d})]}{\sum_{i=1}^{N} \exp[\text{AVG}\,(H_{i,d})]} \tag{5}$$

where $H_{n,d}$ denotes the dth feature of the nth point in the high-level feature. The transpose of the weighted global high-level vector is multiplied by the transpose of the low-level feature $L_{n,d}$. The output represents the response of low-level features to high-level features in each point position. Note that the more similar the feature representations of the two positions are, the higher is their correlation. The weighted low-level feature is fed into an 1×1 convolution layer to reduce the channels, denoted as $D_{n,d}$. The fused feature $E_{n,d}$ is obtained by an element-wise sum between $D_{n,d}$ and the original high-level features $H_{n,d}$.

The weighted low-level features retain detailed information and suppress features that are deemed to be irrelevant or redundant based on high-level semantics.

PointSIFT-GPA Network. With the GPA module, we developed the PointSIFT-GPA network. The input of the architecture are 5-dim vectors (x, y, z, i, r) of 8192 points, which represent the point coordinates (x, y, z), intensity, and return number, respectively. In the downsampling phases, the features extracted from the input point set are processed by a MLP. The point set is shrunk to $1024, 256, 64$ gradually by three consecutive downsampling operations, which contain SA with MSG [28] to capture multiscale features. Features at different scales are concatenated to form multiscale features. For the upsampling part, the FP module proposed in [40] is used for dense features and predictions. The point set is gradually lifted to $256, 1024, 8192$

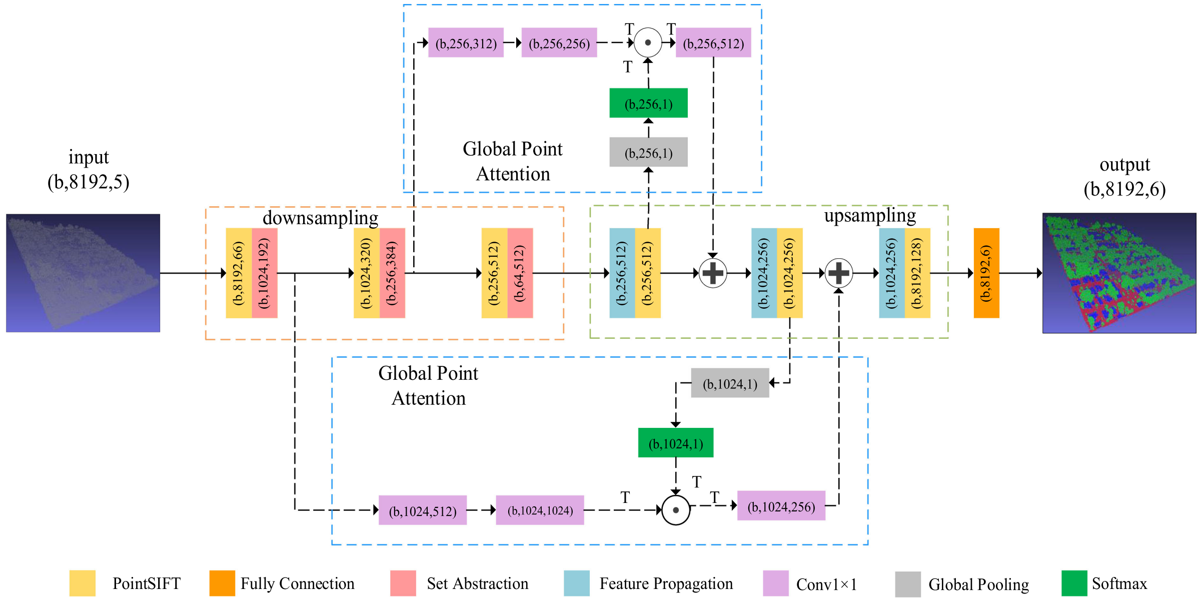


Fig. 10. Illustration of our network architecture. The network consists of downsampling, upsampling, and GPA structures. Both SA and FP modules are introduced in [40]. The PointSIFT module is introduced in [32].

points by three FP layers. Following[28], the PointSIFT module is inserted into all adjacent SA-MSG and FP layers. Point features of the last upsampling layer pass through a fully connected layer for semantic label prediction.

The PointSIFT module is a feature descriptor that describes information in multiple directions and is invariant to scale. The GPA module is inserted between the corresponding SA-MSG and FP layers. The two inputs of GPA are the low-level features of the SA-MSG module and the high-level features of the PointSIFT module.

Data Preprocessing. To train our network, we partition each scene into blocks containing less than K_m points ($K_m = 2^{16}$ in our experiments). For each block, $P_c(\cdot)$ denotes the proportion of each class. To ease the imbalance among the categories, if one block meets the condition that

$$P_c\,(\text{bridge}) + P_c\,(\text{water}) > P_m \tag{6}$$

where P_m is the mean of $P_c(\text{bridge})$ and $P_c(\text{water})$ multiplied by a constant, we randomly rotate this block to create eight new blocks as training data.

From the training data created by cutting and rotation, we dropped all blocks for which $P_c\,(\text{ground}) > P_{\max}$ (with $P_{\max} = 0.85$ in our experiments), which removed all blocks only containing ground points.

These operations created training data with a better balanced class distribution. During inference, we partitioned the point cloud into blocks and joined the results of all blocks.

Dual-Max Cross-Entropy. To guarantee adequate training with minority classes such as water and bridge, we developed a new loss function named dual-max cross-entropy, which increases the influence of points belonging to minority classes. Cross-entropy loss $H(y', y)$ is one of the most popular methods used in deep learning and defined as

$$H(y', y) = -\sum_i W_{y_i} {y_i}' \log(y_i) \tag{7}$$

where y' is the reference value, y is the predicted value normalized by softmax function, and W_{y_i} is the coefficient of y_i.

We define a new coefficient W_{y_i} that depends on the larger of the coefficients of the reference y' and predicted value y and gives the points of minority classes a larger influence on the model during training. The new cross-entropy is described by (8), where W_{y_i} is calculated using (9)

$$H(y', y) = -\sum_i \max\left(W_{y_i}, W_{y_i'}\right) {y_i}' \log\left(y_i\right) \tag{8}$$

$$W_{y_i} = \frac{1}{\text{softmax}(P_{y_i})} \tag{9}$$

where P_{y_i} represent the statistical probability of ith class.

Adding only this weight to the traditional cross-entropy increases the performance only a little. While more points of minority classes are correctly classified, points of majority classes tend to be labeled as a minority class. We applied dual-max cross-entropy to handle these problems, which have the same coefficients. Compared with the traditional loss function, our loss function helps to train a more refined model that is better able to precisely recognize the minority points.

Result Revise Algorithm. Both in training and testing, 512 × 512 m scenes are cut into small blocks. Then, general data processing such as normalization is applied to these blocks before training. This causes the loss of local spatial context information. For example, we usually observed many points around the center of a roof labeled as ground because if the roof is very large a single block can only partially cover it. Even complex models struggle to differentiate this block from ground

Algorithm 1: Result Revise Algorithm.

```
for $i = 1$ to $N$do
  Use KNN to get $k_{i1}$ ... $k_{iK}$;
end for
for $i = 1$ to $N$do
  if $i_{th}$ point labeled as buildingthen
    $T, L =$ DFS($i$);
    if $T > T_p$ then
      for $j = 1$ to $T$do
        Label $(L_j)_{th}$ point as building;
      end for
    end if
  end if
end for
```

Algorithm 2: Deep First Search (DFS).

```
Require:
  The point index, $i$;
Ensure
  The number of searched points, $T$;
  The searched points list, $L$;
$T \Leftarrow 0$;
$L \Leftarrow \{\}$;
for $j = 1$ to $K$do
  if $Z_{k_{ij}} - Z_i < \Delta Z$ then
    Put $k_{ij}$ in $L$;
    $T \Leftarrow T + 1$;
    DFS($k_{ij}$);
  end if
end for
Return $T$, $L$;
```

points after normalization. To solve this problem, we developed an algorithm based on depth-first search (DFS) and kNN. In our algorithm, kNN is used to create the search targets for every point and DFS is used to revise all incorrectly labeled points. The complete revise algorithm is shown in Algorithm 1 and the DFS part is shown in Algorithm 2.

First, we use kNN to find the k nearest points for each point in a scene and save them in a list K where k_{ij} represent the j_{th} nearest point of the i_{th} point. Then, for each point labeled as building in the original result, we use DFS to search all of its k nearest points. If this point and one of its nearest points are extremely close in height (i.e., $Z_{k_{ij}} - Z_i < \Delta Z$, with $\Delta Z = 10^{-1}$), we suppose they belong to the same class. The DFS records all the searched points in a list L and the total number of them as T. Before revising all points in L as building, we first check whether their number T is greater than a threshold T_p. If not, these points are insufficient to constitute a large building and changing their classes is not necessary. This step is very important as it avoids the influence of an incorrect classification on the original result.

In our experiments , all buildings mistaken for ground were correctly revised.

TABLE IV
SEMANTIC SEGMENTATION RESULTS AND MODEL SIZE ON US3D TEST SET. THE DFS DENOTES OUR POSTPROCESSING ALGORITHM RESULT REVISE ALGORITHM (RRA)

Network	mIoU	OA	Parameters
PointSIFT	0.910	0.976	14.3M
PointSIFT+MSG	0.922	0.980	13.7M
PointSIFT+MSG+GPA	0.935	0.985	14.7M
PointSIFT+MSG+GPA+RRA	0.945	0.986	–

B. Results and Discussion

Experimental Setting. All models are implemented in Tensorflow and run on a NVIDIA GeForce GTX 1080Ti GPU using the Adam optimizer. The learning rate was initially set to 0.001. ReLU and batch normalization were applied after each layer. For each model, we used the farthest point sampling method [28] to select the point set, and the number of initial sampling points was 1024.

Segmentation Results. Fig. 11 shows a comparison between PointSIFT-GPA, PointNet [40], PointNet++ [28], PointSIFT [32], Point2sequence[41], PointConv [33], and PointCNN[36]. For fair comparison, all results in Fig. 11 were obtained under the same conditions. The curves describe OA and mIoU during the training phase. PointSIFT-GPA achieved the best overall performance. However, it is worth noting that, compared with the other methods, PointConv [33] reached higher OA and mIoU values in the initial epochs because of the large number of network parameters and fast feature fitting. PointSIFT-GPA had a faster convergence rate than the baseline network PointSIFT. Moreover, OA and mIoU values were higher than those for the baseline network after convergence.

Ablation Studies. For a better understanding of PointSIFT-GPA, we conducted experiments to evaluate the improvements of its individual components. Table IV shows the experimental results, where PointSIFT+MSG denotes the PointSIFT network with the MSG module and PointSIFT+MSG+GPA represents the PointSIFT network with the GPA and MSG modules. The GPA module improved the network performance significantly. Moreover, the network convergence rate was slightly increased and the oscillation of the OA curve was small. This implies that the GPA module can effectively extract contextual information from local regions, making the model more robust and more adaptive to the noise between different batches. PointSIFT+MSG achieved an mIoU of 0.922, which represents an improvement of 0.012. The MSG module was proposed to fuse multiscale features for extracting more detailed local point cloud features. Furthermore, employing the GPA and MSG modules outperformed the PointSIFT with the MSG module by 0.014. The mIoU is 0.935 with an 1.6 M increment on parameters. With the proposed postprocessing method (RRA), the final performance reached an mIoU of 0.945 and had an accuracy of 0.986.

Fig. 12 shows qualitative results using the PointSIFT-GPA method with different modules. Comparing the first and second

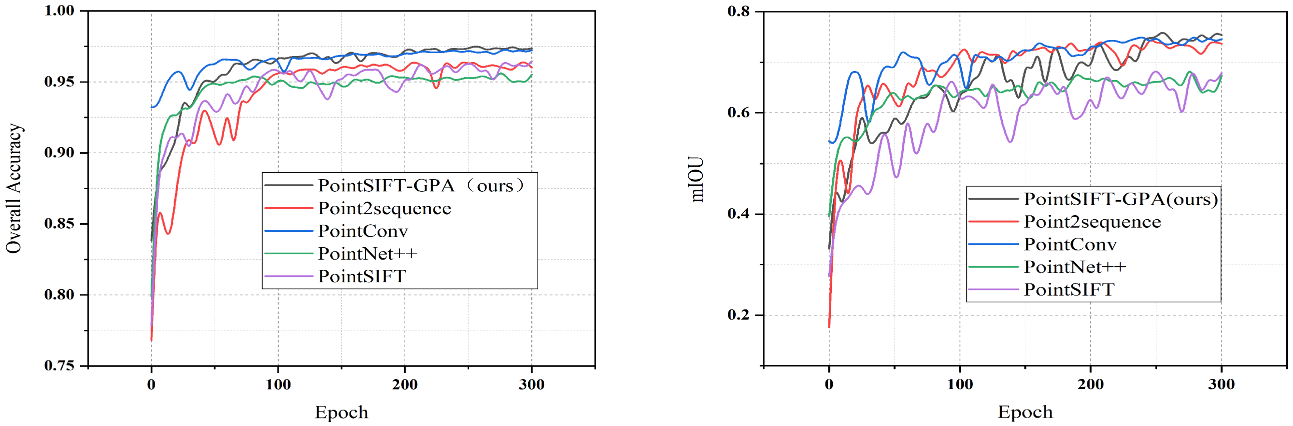


Fig. 11. Curves show the overall accuracy and mean IoU of PointSIFT-GPA with state-of-the-art methods for semantic segmentation of the US3D dataset.

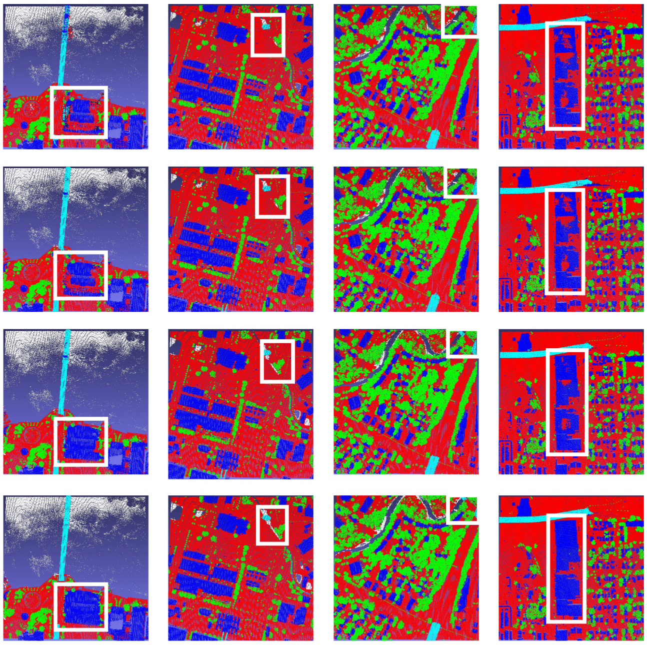


Fig. 12. Visual improvement on the test set of US3D [31]. From top to bottom: the result of PointSIFT, PointSIFT+MSG, PointSIFT+MSG+GPA, PointSIFT+MSG+GPA+RRA. From left to right: JAX-213, JAX-119, JAX-327, OMA-272. Red represents ground, blue represents building, green represents high vegetation, white represents water, cyan-blue represents elevated road. The white box is the obvious difference between models.

rows, it can be seen that PointSIFT with MSG identifies more detailed building points than PointSIFT. MSG performs dense sampling with multiscale patterns in local areas to capture multiscale features. It also shows that PointSIFT + MSG + GPA further identifies more ground points. The network guides the high-level features to weight low-level features. Thus, the model not only recognizes high-level semantic information, but also retains detailed low-level information.

Model Complexity. Table V reports the time and space cost of several state-of-the-art methods and PointSIFT-GPA. The forward time is recorded with a batch size of 6 and 8192 input points. All experimental results are recorded under the same hardware environment. The PointSIFT-GPA has 14.7 M parameters and runs on a GPU at 399 s per batch for training/inference. As shown in Table V, compared to PointSIFT, the PointSIFT-GPA increased the number of parameters by 1.6 M. It is worth noting that for PointSIFT-GPA, while its model is slightly larger than PointSIFT, the accuracy is significantly improved due to the use of context information.

TABLE V
COMPLEXITY, FORWARD TIME, AND ACCURACY OF DIFFERENT MODELS IN US3D TEST SET

Methods	Model Size (M)	Accuracy	Forward time (ms)
PointSIFT	57.2	0.910	451
Point2Sequence	7.36	0.870	2230
Pointnet++	3.88	0.865	339
PointCNN	46	0.906	214
PointConv	86.8	0.897	6480
PointSIFT-GPA	58.8	0.935	499

VI. CONCLUSION

For a long time point clouds, in particular those acquired by LiDAR, were regarded as a stand-alone product and were used to measure purely geometric properties of a scene, including height, changes in height, vegetation volume, etc. The first automatic procedures to analyze point cloud data focused on grouping points into meaningful parts based on their geometric properties such as proximity, surface normals, and curvature. However, in recent years, the semantic analysis of point clouds has become a research focus both within academia and industry. This has prompted significant progress in hardware (e.g., affordable and light-weight 3-D systems such as portable laser scanners and time-of-flight cameras such as the MS Kinect for close-range scenes) and software development, mainly for the extension of deep Learning methods to 3-D data. Modern approaches are able to provide semantically annotated 3-D models that are of great importance in a wide range of applications including urban planning and monitoring of natural environments.

The 2019 Data Fusion Contest of the Image Analysis and Data Fusion (IADF) Technical Committee of the IEEE Geoscience and Remote Sensing Society addressed the challenges within the context of semantic segmentation of 3-D point clouds by providing LiDAR based 3-D data as well as semantic annotations (and—beyond the Track 4 described in this article—also high-resolution image data) for two different sites resulting in 69 data tiles of more than 320 GB. This allows for efficient benchmarking of methods that aim to solve large-scale semantic 3-D annotation tasks.

Despite the challenges of this contest (e.g., the large amount of data), participation increased compared with previous years [19]. With a total of 45 participating countries and winning approaches in all tracks from China, Germany, Nepal, and the USA, DFC19 was a truly global event. All winners of Track 4 used PointNet-based networks for the semantic analysis of the data. The top two solutions improved upon the PointSIFT network architecture with mechanisms to combine multiscale features. Various methods for task-specific postprocessing of model outputs as well as the use of ensembles of multiple predictors also allowed for significant performance gains.

After the contest, the data remained accessible for further research on the globally accessible data platform IEEE DataPort[1] and the evaluation servers were reopened and made accessible from the contest website.[2] While addressing semantic 3-D at such a scale was unprecedented, many promising improvements can already be foreseen. In addition to scaling up (with more scenes and more semantic classes—including objects and underrepresented land-use classes), new problems can also be addressed, e.g., the evolution along the time dimension (temporal changes in semantics and in 3-D). Moreover, it will be crucial to investigate how currently emerging machine-learning techniques such as weak supervision and self-supervised learning can be harnessed to build well performing models.

Acknowledgment

The authors would like to thank K. Foster, G. Christie, M. Bosch, S. Almes, and S. Wang for their contributions in preparing the dataset and baseline code.

Preparation and public release of the challenge data and code was supported by the Intelligence Advanced Research Projects Activity (IARPA). Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA or the U.S. Government.

References

[1] D. Poli and T. Toutin, "Review of developments in geometric modelling for high resolution satellite pushbroom sensors," *The Photogrammetric Rec.*, vol. 27, pp. 58–73, 2012.

[2] A. Wehr and U. Lohr, "Airborne laser scanning—An introduction and overview," *ISPRS J. Photogrammetry Remote Sens.*, vol. 54, no. 2, pp. 68–82, 1999.

[3] G. Vosselman and H. Maas, Eds., *Airborne and Terrestrial Laser Scanning*. United Kingdom: CRC Press, 2010.

[4] P. Axelsson, "Processing of laser scanner data-algorithms and applications," *ISPRS J. Photogrammetry Remote Sens.*, vol. 54, no. 2, pp. 138–147, 1999.

[5] W. Y. Yan, A. Shaker, and N. El-Ashmawy, "Urban land cover classification using airborne LiDAR data: A review," *Remote Sens. Environ.*, vol. 158, pp. 295–310, 2015.

[6] Y. Xie, J. Tian, and X. X. Zhu, "A review of point cloud semantic segmentation,," to be published, doi: 10.1109/MGRS.2019.2937630.

[7] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. M. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3012–3021, Oct. 2007.

[8] F. Pacifici, F. Del Frate, W. J. Emery, P. Gamba, and J. Chanussot, "Urban mapping using coarse SAR and optical data: Outcome of the 2007 GRSS Data Fusion Contest," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 3, pp. 331–335, Jul. 2008.

[9] G. Licciardi *et al.*, "Decision fusion for the classification of hyperspectral data: Outcome of the 2008 GRS-S data fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3857–3865, Nov. 2009.

[10] N. Longbotham *et al.*, "Multi-modal change detection, application to the detection of flooded areas: Outcome of the 2009–2010 Data Fusion Contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 331–342, Feb. 2012.

[11] F. Pacifici and Q. Du, "Foreword to the special issue on optical multiangular data exploitation and outcome of the 2011 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 3–7, Feb. 2012.

[12] C. Berger *et al.*, "Multi-modal and multi-temporal data fusion: Outcome of the 2012 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1324–1340, Jun. 2013.

[13] C. Debes *et al.*, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.

[14] W. Liao *et al.*, "Processing of multiresolution thermal hyperspectral and digital color data: Outcome of the 2014 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2984–2996, Jun. 2015.

[15] M. Campos-Taberner *et al.*, "Processing of extremely high resolution LiDAR and RGB data: Outcome of the 2015 IEEE GRSS data fusion contest. Part A: 2D contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 12, pp. 5547–5559, 2016.

[16] A.-V. Vo *et al.*, "Processing of extremely high resolution LiDAR and RGB data: Outcome of the 2015 IEEE GRSS data fusion contest. Part B: 3D contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 12, pp. 5560–5575, Dec. 2016.

[17] L. Mou *et al.*, "Multi-temporal very high resolution from space: Outcome of the 2016 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3435–3447, Aug. 2017.

[18] N. Yokoya *et al.*, "Open data for global multimodal land use classification: Outcome of the 2017 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1363–1377, May 2018.

[19] Y. Xu *et al.*, "Advanced multi-sensor optical remote sensing for urban land use and land cover classification: Outcome of the 2018 IEEE GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1709–1724, Jun. 2019.

[20] M. Bosch, K. Foster, G. Christie, S. Wang, G. D. Hager, and M. Brown, "Semantic stereo for incidental satellite images," in *Proc.Winter Conf. Appl. Comput. Vision*, 2019, pp. 1524–1532.

[21] S. Kunwar *et al.* "Large-scale semantic 3D reconstruction: Outcome of the 2019 IEEE GRSS data fusion contest - Part A," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, to be published, doi: 10.1109/JSTARS.2020.3032221.

[22] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.net: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inform. Sciences*, vol. IV-1/W1, pp. 91–98, May 2017.

[23] B. Le Saux, N. Yokoya, R. Hänsch, and M. Brown, "IEEE Dataport: Data fusion contest 2019," 2019. [Online]. Available: http://dx.doi.org/10.21227/c6tm-vw12

[1] [Online]. Available: https://ieee-dataport.org/open-access/data-fusion-contest-2019-dfc2019

[2] [Online]. Available: http://www.grss-ieee.org/community/technical-committees/data-fusion/2019-ieee-grss-data-fusion-contest/

[24] K. Foster, G. Christie, and M. Brown, "IEEE Dataport: Urban semantic 3D dataset," 2020. [Online]. Available: http://dx.doi.org/10.21227/9frn-7208

[25] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of LiDAR data and building object detection in urban areas," *ISPRS J. Photogrammetry Remote Sens.*, vol. 87, pp. 152–165, Jan. 2014.

[26] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, "TerraMobilita/iQmulus urban point cloud analysis benchmark," *Comput. Graph.*, vol. 49, pp. 126–133, Jun. 2015.

[27] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *Int. J. Robot. Res.*, vol. 37, no. 6, pp. 545–557, 2018.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 5099–5108.

[29] "GitHub: Data fusion contest 2019." [Online]. Available: https://github.com/pubgeo/dfc2019

[30] Y. Lian, T. Feng, and J. Zhou, "A dense Pointnet++ architecture for 3D point cloud semantic segmentation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Yokohama, Japan, 2019, pp. 5061–5064.

[31] M. Jia, A. Li, and Z. Wu, "A global Point-Sift attention network for 3D point cloud semantic segmentation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.* , Yokohama, Japan, 2019, pp. 5065–5068.

[32] M. Jiang, Y. Wu, and C. Lu, "Pointsift: A sift-like network module for 3D point cloud semantic segmentation," *CoRR*, vol. abs/1807.00652, 2018. [Online]. Available: http://arxiv.org/abs/1807.00652

[33] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 9621–9630.

[34] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUS)," in *Proc. 4th Int. Conf. Learn. Representations, {ICLR} 2016*, San Juan, Puerto Rico, May 2-4, 2016.

[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2014.

[36] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H.Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. New York, NY, USA: Curran Associates, Inc., 2018, pp. 820–830.

[37] W. Bo, L. Yang, L. Bo, and H. Lei, "DGCNN: Disordered graph convolutional neural network based on the Gaussian mixture model," *Neurocomputing*, vol. 321, pp. 346–356, 2017.

[38] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018. [Online]. Available: http://arxiv.org/abs/1805.10180

[39] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jul. 2017, pp. 77–85.

[41] X. Liu, Z. Han, Y. Liu, and M. Zwicker, "Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," 2018. [Online]. Available: http://arxiv.org/abs/1811.02565