3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation

Dening Lu, Kyle Gao, *Graduate Student Member, IEEE*, Qian Xie,
Linlin Xu, *Member, IEEE*, and Jonathan Li, *Fellow, IEEE*

***Abstract*— Although the application of Transformers to 3-D point cloud processing has achieved significant progress and success, it is still challenging for existing 3-D Transformer methods to efficiently and accurately learn both valuable global and local features for improved applications. This article presents a novel point cloud representational learning network, called 3-D Dual Self-attention global local (GLocal) Transformer Network (3DGTN), for improved feature learning in both classification and segmentation tasks, with the following key contributions. First, a GLocal feature learning (GFL) block with the dual self-attention mechanism [i.e., a novel point-patch self-attention, called PPSA, and a channel-wise self-attention (CSA)] is designed to efficiently learn the global and local context information. Second, the GFL block is integrated with a multiscale Graph Convolution-based local feature aggregation (LFA) block, leading to a GLocal information extraction module that can efficiently capture critical information. Third, a series of GLocal modules are used to construct a new hierarchical encoder–decoder structure to enable the learning of information in different scales in a hierarchical manner. The proposed framework is evaluated on both classification and segmentation datasets, demonstrating that the proposed method is capable of outperforming many state-of-the-art methods on both synthetic and LiDAR data. *Our code has been released at https://github.com/d62lu/3DGTN*.**

***Index Terms*— Graph convolution, LiDAR data processing, point cloud classification, point cloud segmentation, self-attention mechanism, transformer.**

I. INTRODUCTION

POINT cloud classification and segmentation are fundamental tasks in 3-D computer vision. Point clouds, being flexible, simple, and with easy-to-use data structures, are commonly used in 3-D mapping, robotics, autonomous navigation, and city information modeling. From the perspective of point cloud processing, both local and global features play an important role in classification and segmentation tasks. Local features refer to the features that capture the local geometric patterns and details of the point cloud. Global features refer to the features that capture the overall shape and structure of the entire point cloud. A combination of global and local features (called *GLocal* features here) is able to provide the model with a more complete representation of the target point cloud.

For classification and segmentation tasks, many types of deep learning architectures have been experimented with in the recent past. Among these, the Transformer [1] architecture emerged as a powerful point cloud feature extraction backbone, performing exceedingly well on LiDAR point cloud classification and segmentation. [2], [3], [4], [5]. First developed for natural language processing, the Transformer is a low-inductive bias network that is capable of learning long-range features. Since then, Transformers have successfully been applied to 2-D and 3-D computer vision to various tasks, achieving state-of-the-art results across a wide variety of benchmarks.

Although existing 3-D Transformer approaches demonstrated strong feature learning capabilities in 3-D point cloud applications, they still have limitations in terms of modeling both the local information and global information in an efficient and accurate manner. This article presents a 3-D Dual-attention GLocal Transformer Network, called 3DGTN. It focuses on addressing the difficulty of effectively exploiting global and local features for point cloud classification and segmentation. Many current Transformer methods either emphasize local information extraction or struggle to integrate global and local features accurately. The proposed point-patch self-attention (PPSA) mechanism in 3DGTN aims to overcome this limitation. 3DGTN is tailor-designed to improve combined global and local feature learning in 3-D point cloud data processing, with the following key characteristics.

1) A GLocal feature learning (GFL) block with the dual self-attention mechanism is designed to efficiently learn the GLocal context information. The PPSA approach can better capture global correlation among local neighborhoods. The dual-attention mechanism integrates PPSA and channel-wise self-attention (CSA) to improve the learning of critical information in both the spatial domain and feature domain.

Manuscript received 17 August 2023; revised 4 December 2023 and 28 February 2024; accepted 31 March 2024. Date of publication 25 April 2024; date of current version 14 May 2024. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2019-06744. The work of Dening Lu was supported by China Scholarship Council under the Ph.D. Scholarship under Grant 202106830030. *(Corresponding authors: Linlin Xu; Jonathan Li.)*

Dening Lu, Kyle Gao, and Linlin Xu are with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: d62lu@uwaterloo.ca; y56gao@uwaterloo.ca; l44xu@uwaterloo.ca).

Qian Xie is with the Department of Computer Science, University of Oxford, OX1 3QD Oxford, U.K. (e-mail: qian.xie@cs.ox.ac.uk).

Jonathan Li is with the Department of Systems Design Engineering and the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Digital Object Identifier 10.1109/TGRS.2024.3393845

2) The GFL block is integrated with a local feature aggregation (LFA) block into a GLocal information extraction module to enable the learning of both valuable global information and critical local information. The LFA block is designed based on the graph convolution network (GCN) to improve both the efficiency and accuracy of local information extraction.
3) The GLocal modules are used to construct a new hierarchical encoder–decoder structure to enable the learning of information at different scales in a hierarchical manner, leading to a general point cloud representation network that can improve both classification and segmentation.

Extensive experiments comparing the proposed approach with many state-of-the-art algorithms on many datasets, i.e., ModelNet40, ScanObjectNN, ShapeNet, and Titan MultiSpectral (MS) LiDAR datasets, demonstrate that our method exceeds previous state-of-the-art performance in both classification and segmentation tasks.

II. Related Work

Transformer-based methods tailored for point cloud data can be broadly categorized into two main groups: global Transformer-based methods and local Transformer-based methods. Here, we review existing approaches in both categories and summarize the limitations.

A. Global Transformers in 3-D Point Cloud Processing

The global Transformer approaches focus on learning large-scale context information from the 3-D point cloud to improve classification and segmentation. point cloud transformer (PCT), as a standard global Transformer network, was proposed in [6]. In PCT, all input points were leveraged for global feature extraction. PCT first adopted a neighborhood-embedding strategy to aggregate the local information, followed by feeding the embedded features into four stacked global Transformer blocks. At last, it utilized a global max and average (MA) pooling to extract the global information for point cloud classification. The segmentation network variant of PCT [6] had the same feature encoding backbone as the classification network variant. However, the decoder first concatenated the pooled global feature with each point feature, enhancing the perception of global information for each point. Then, the concatenated features were fed into a series of MLP layers for dense prediction, following PointNet [7].

3CROSSNet proposed in [8] used multiscale global information for classification. Taking the raw point cloud as input, it first generated three point subsets with different resolutions using Farthest Point Sampling (FPS). Second, it established k-nearest neighborhood (kNN) [7] and extracted local information using a series of multilayer perception (MLP) modules for each point subset. Third, the cascaded global Transformer blocks were applied to extract the global information of each subset. At last, given the multiscale global features, 3CROSSNet used the cross-level cross-attention (CLCA) and cross-scale cross-attention (CSCA) modules to capture long-range inter- and intralevel dependencies for classification.

Instead of using raw point clouds, Stratified Transformer [9] took 3-D voxels as input to the segmentation network. It applied Transformer blocks in predefined local windows, following Swin Transformer [10]. To capture the global information and establish connections between different windows, it presented a novel key sampling strategy, enlarging the effective receptive field for each query point.

B. Local Transformers in 3-D Point Cloud Processing

As a local Transformer network, point transformer (PT) [11] focused on extracting local information by the Transformer. A downsampled pointset was passed through five local Transformer blocks. Specifically, for each block, PT used kNN for sampling points, and then utilized a vector-attention mechanism to capture local features. After five local Transformer blocks, PT used a global MA pooling to extract the global feature for classification. Local feature transformer network (LFT-Net) [12] had a similar architecture. However, it used an additional trans-pooling module to alleviate the feature loss during the pooling. For 3-D point cloud segmentation, PT [11] developed the segmentation network based on its classification framework. The authors designed a U-net-style architecture for segmentation, where the decoder was symmetric to the encoder. Since it used a hierarchical structure in the encoder, a transition-up module with trilinear interpolation was proposed in the decoder for point cloud upsampling.

C. Limitations of Current 3-D Transformer-Based Networks

Despite the great success of Transformers in point cloud classification and segmentation, existing 3-D Transformer methods tend to only consider local information extraction or struggle to learn both global and local features effectively. This issue makes it still challenging for 3-D Transformer methods to capture the global information of the target accurately while preserving the local features. For example, PT [11] only utilized Transformer blocks in local neighborhoods, while ignoring global feature learning. FlatFormer [13] used Transformer blocks to extract window-based local features, and designed a window shift strategy to indirectly achieve global feature learning. PCT [6] only captured local information at the beginning of the network, as data preprocessing. It cannot dynamically fuse the local information with the global information extracted from each stage in the network. Recently, there have been several works [14], [15], [16], [17], [18], [19] that extract both local and global features in a simple cascading way. PatchFromer [17], SPFormer [18], and SPT [19] all used Transformer blocks to capture global features from aggregated superpoint-based local features. However, it is easy for them to lose local neighborhood information. Therefore, this article proposes a novel PPSA (Section III-C) mechanism to improve global and local feature learning. It aims to explicitly fuse the local neighborhood and global information of the target. To the best of our knowledge, our 3DGTN is the first work to introduce combined GLocal feature learning to 3-D Transformers for point cloud processing.

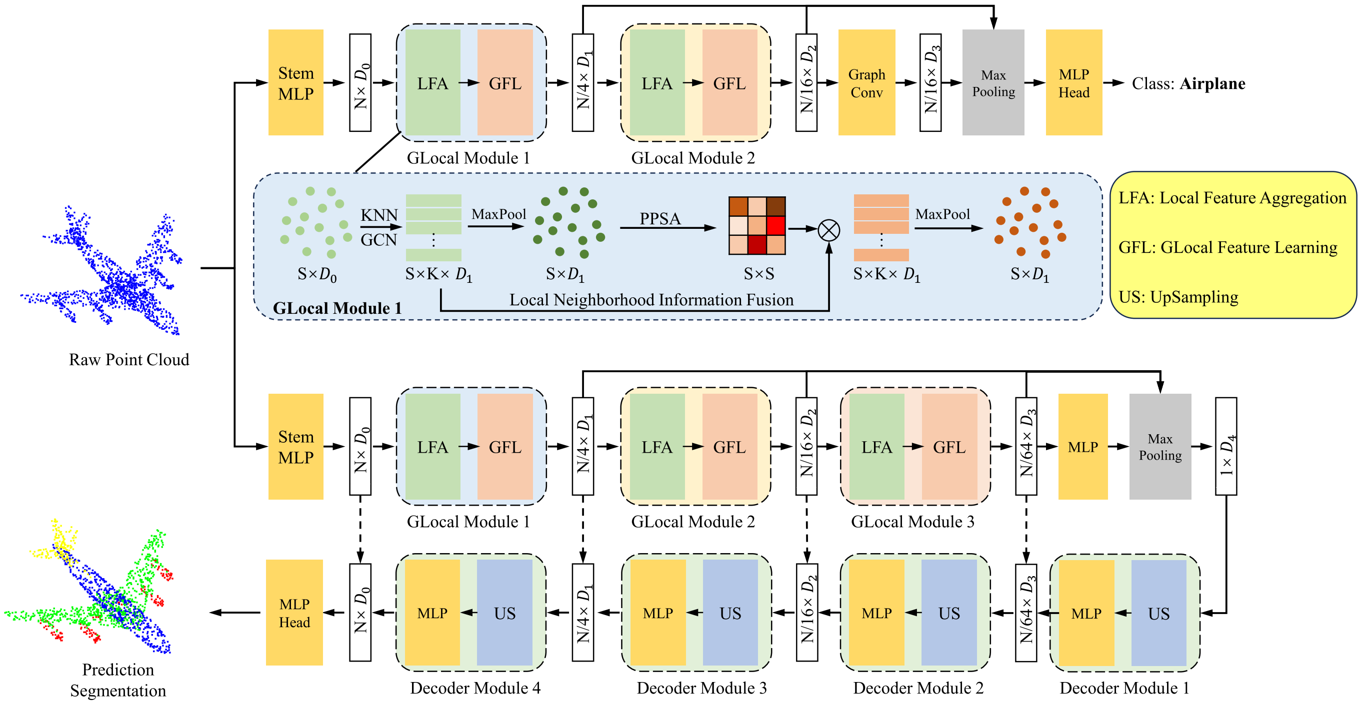


Fig. 1. (Top) 3DGTN networks for point cloud classification. (Bottom) segmentation, where GCN-based LFA blocks and dual self-attention-based GFL blocks are designed for strong feature representation. A brief illustration of the GLcoal Module 1 is provided to highlight GLocal feature generation. Please refer to Fig. 2, Sections III-B and III-C for more details.

III. THREE-DIMENSIONAL DUAL-ATTENTION GLOCAL TRANSFORMER

In this section, we introduce the encoder and decoder structures of our 3DGTN for both point cloud classification and segmentation. We first show the pipeline of our method, then introduce the main blocks in the encoder and decoder, respectively.

A. Overview

Fig. 1 shows the overall pipeline of our method. Our classification and segmentation networks use the same encoder architecture. After that, the classification network utilizes an MLP head to obtain the final classification results, while the segmentation network utilizes a decoder with trilinear interpolation-based upsampling for dense prediction.

The original point cloud is taken as input to the encoder. We first design a stem MLP block to project the input data into a higher dimensional space. After that, the projected features are fed into stacked LFA and GFL blocks in a hierarchical manner for GLocal feature extraction. Specifically, the LFA block is adapted from the multiscale GCN [20], and the GFL block is adapted from the Transformer. Following this, we use the max-pooling operation on the output feature maps of each module, to obtain the GLocal feature of each level. Then, we concatenate them for multilevel GLocal feature generation. Given the extracted feature, we leverage an MLP head for the point cloud classification task, which consists of two fully connected layers with batch normalization and RELU activation. For the segmentation task, the extracted features are then taken as input to the decoder. To improve efficiency, we adopt an ALL-MLP decoder structure, instead of a symmetric one. In the upsampling block, the interpolated points are concatenated with the corresponding feature points from the encoder via a skip connection. The trilinear upsampling method we used has been widely applied to hierarchical networks of point cloud processing. It generates new points by considering the weighted averages of neighboring points in the geometric space as 3-D linear interpolation, providing an effective method to enhance the density and precision of 3-D data representations. We note that the number of modules in the encoder and decoder can vary according to the number of input points. In our experiments, we designed a two-module encoder for the classification task (1024 points), but a three-module encoder and corresponding decoder for the segmentation task (2048 points).

B. LFA Block

We adopt the GCN-based LFA block for LFA. The LFA block (Fig. 2) is introduced as follows.

The input point cloud is first downsampled to $N/4$ points via FPS, generating a sampled point subset S, where N is the number of the input points. After that, the LFA block constructs multiscale k-NN neighborhoods (three scales k_1, k_2, k_3 in our experiments) for each sampled point, to ensure the diversity of the receptive fields. In each neighborhood χ_i of the sampled point S_i, a fused feature $\mathbb{C}_{ij}$ is generated by computing the difference between the jth neighborhood point χ_{ij} in χ_i and S_i

$$\mathbb{C}_{ij} = \text{concat}(\mathbb{F}_{ij} - \mathbb{F}_i, \mathbb{F}_i) \tag{1}$$

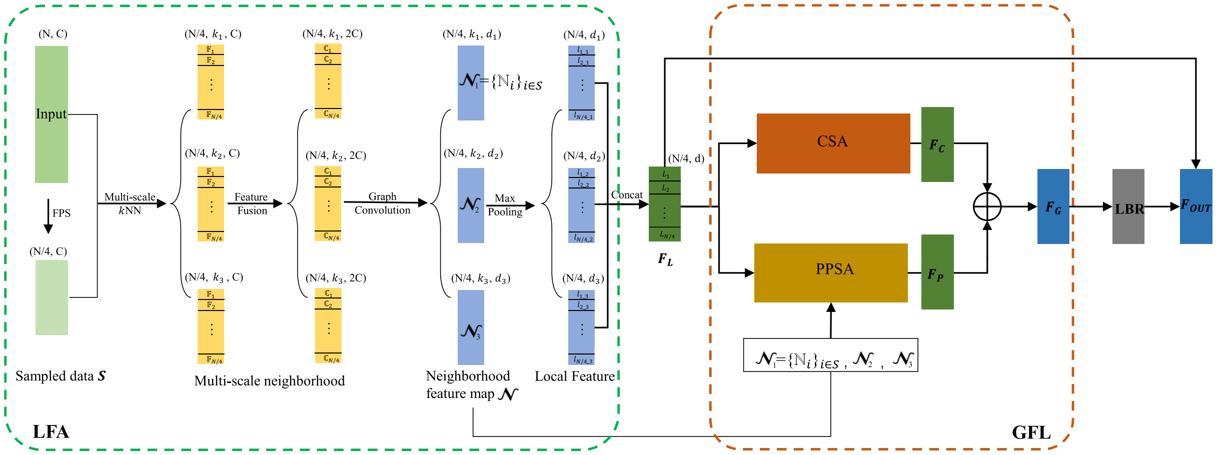


Fig. 2. Architecture of GLocal Module 1, which consists of an LFA block and a GFL block.

where $\mathbb{F}_i$ and $\mathbb{F}_{ij}$ represents the feature of S_i and χ_{ij}, respectively. Given the fused neighborhood feature, the Graph Convolution in χ_i can be formulated as

$$l_i = \underset{j \in \chi_i}{\text{maxpooling}} \left(\text{Conv}(\mathbb{C}_{ij})\right) \tag{2}$$

where l_i is the aggregated local feature of S_i, Conv is a convolution operation with 1 × 1 kernels. Specifically, in Fig. 2, we denote the dimension of the input feature map as (N, C), and the dimension of $\mathbb{C}_{ij}$ is $2C$. Furthermore, we define different output dimensions of Graph Convolution for different scale neighborhoods: d_1, d_2, and d_3, where $d_1 < d_2 < d_3$ $(k_1 < k_2 < k_3)$. $\text{Conv}(\mathbb{C}_{ij})$ establishes semantic relationships between the sampling point S_i and neighborhood point χ_{ij}. As such, a neighborhood feature set containing local information, $\mathbb{N}_i$ of S_i, is generated. Then, the max-pooling operation is used to aggregate the local information to S_i.

The multiscale local feature L_i of S_i, can be expressed as via a concatenation as

$$L_i = \text{concat}(l_{i\_1}, l_{i\_2}, l_{i\_3}) \tag{3}$$

where $l_{i\_1}, l_{i\_2}, l_{i\_3}$ represent three local features of S_i at three different scales.

C. GFL Block

Our GFL block contains two kinds of self-attention mechanisms: PPSA and CSA. PPSA, as a novel point-wise self-attention mechanism, is proposed to fuse the global features and local neighborhood information extracted from the LFA block for better GFL. CSA is utilized to measure the correlation among different feature channels. It is able to improve context information modeling by highlighting the role of interaction across various channels. A detailed introduction to these two mechanisms is as follows.

1) Point-Patch Self-Attention: PPSA fuses local and global features. As shown in Fig. 3, the aggregated features $F_L = \{L_i\}_{i \in s} \in R^{s \times d}$ from the LFA block is taken as input, where s is the number of sampled points in S, and d denotes the feature dimension of F_L. We first project F_L into two different feature spaces to generate Query, Key matrices

$$\begin{aligned} \text{Query} &= F_L W_{QP} \\ \text{Key} &= F_L W_{KP} \end{aligned} \tag{4}$$

where $W_{QP} and W_{KP}$ are learnable weight matrices. Then, the attention map $M_P \in R^{s \times s}$ of PPSA can be formulated as

$$M_P = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right) \tag{5}$$

where Q, K denote the Query, Key matrices, and B is a learnable position encoding matrix defined by [11]. Next, we treat the neighborhood feature map $\mathcal{N} = \{\mathbb{N}_i\}_{i \in s}$ at each scale as the Value branch, instead of F_L used by the vanilla PSA. In other words, the elements in the attention map are taken as weights of the corresponding neighborhood feature sets in $\mathcal{N}$. Then, the output neighborhood feature set is obtained by computing a weighted sum of all input sets. As such, we leverage all the points including sampling points and neighborhood points for the GLocal information extraction, instead of only sampling points. This method is able to improve the feature learning and mitigate the local information loss caused by the pooling operation in 2. Given the aforementioned attention map M_P and the Value matrix, the output GLocal feature can be expressed as

$$F_o = \text{maxpooling}\,(M_P V) \tag{6}$$

where V denotes the Value matrix, i.e., $\mathcal{N}$. The detailed algorithm flow and feature dimension transformation of PPSA are shown in Fig. 3. We note that there are three neighborhood feature maps $\mathcal{N}$ for each sampled point S_i because of the multiscale grouping strategy, which are denoted as $\mathcal{N}_1, \mathcal{N}_2$, and $\mathcal{N}_3$. Correspondingly, we obtain three output GLocal features at different scales, F_{o1}, F_{o2}, and F_{o3}. At lastly, we concatenate them to get the final point-wise GLocal feature F_P

$$F_P = \text{concat}(F_{o1}, F_{o2}, F_{o3}) \tag{7}$$

where $F_{o1}, F_{o2}, and F_{o3}$ are generated from neighborhood feature maps at different scales.

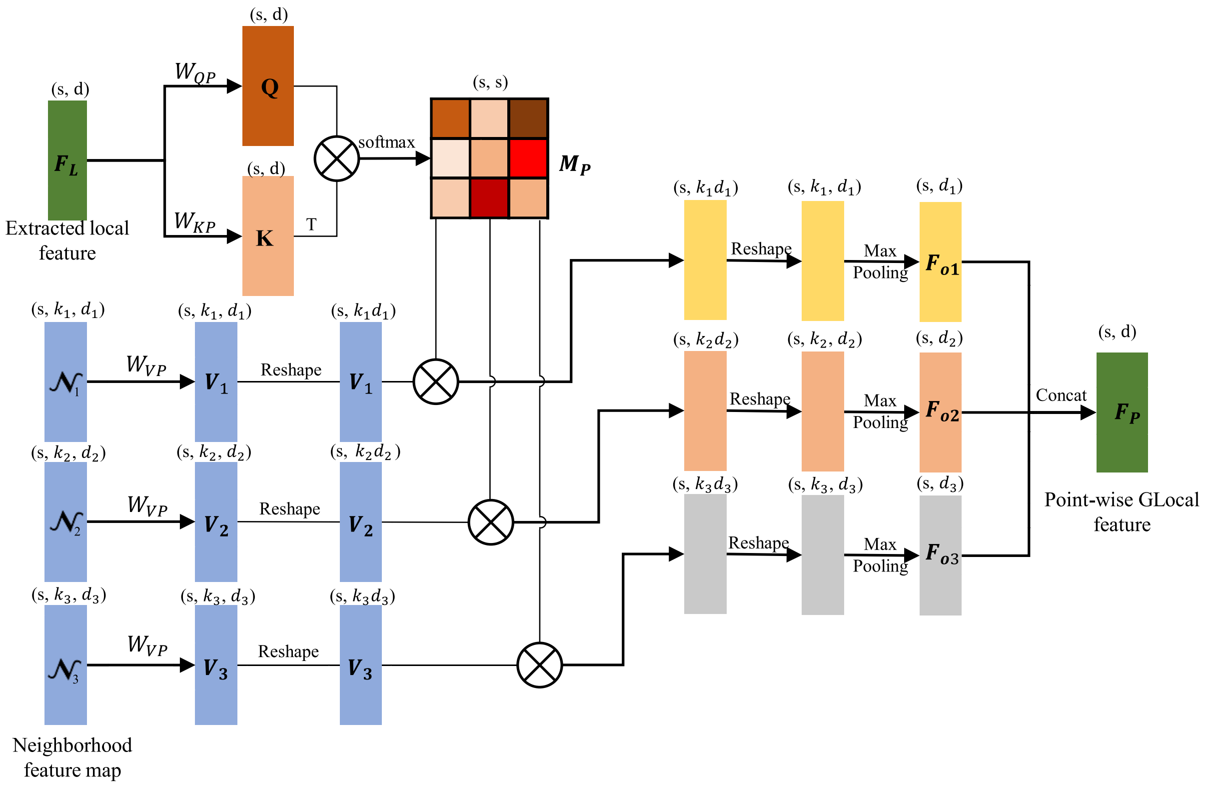


Fig. 3. PPSA mechanism. It utilizes the sampling point features and the corresponding neighborhood (patch) feature maps for point-wise GLocal feature extraction.

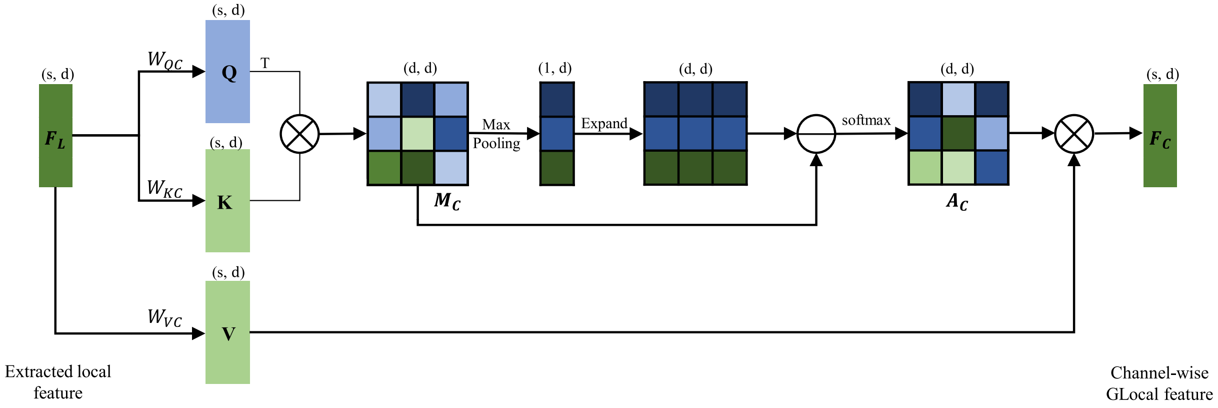


Fig. 4. CSA mechanism. An affinity matrix A_C is designed to avoid aggregating redundant features, enhancing the channel-wise GLocal feature representation.

2) Channel-Wise Self-Attention: Apart from the PPSA mechanism, we also utilize the CSA mechanism to capture context dependencies in the channel dimension. It enables the model to build connections among different channels, allowing it to focus on different feature channels depending on input data. As shown in Fig. 4, given the aggregated local feature $F_L \in R^{s \times d}$, we first compute the attention map $M_C \in R^{d \times d}$ of CSA as

$$M_C = K^T Q = (F_L W_{KC})^T (F_L W_{QC}). \tag{8}$$

where the shapes of K, Q are reduced to $(s/8) \times d$ by weight matrices W_{KC} and W_{QC}, to improve efficiency. Inspired by [21], we calculate the affinity matrix A_C based on M_C, to measure the difference among channels, which can be expressed as

$$A_C = \text{softmax}(\text{expand}(\text{maxpooling}(M_C)) - M_C) \tag{9}$$

where $\text{maxpooling}(M_C) \in R^{d \times 1}$ extracts the maximum value of each row in M_C and $\text{expand}(\cdot)$ expands the matrix $\text{maxpooling}(M_C)$ to the same size as M_C by column repetition. From the subtraction, the larger in magnitude the element in A_C, the lower the similarity of the corresponding two channels. As such, CSA tends to focus on channels with significant differences, avoiding aggregating similar/redundant information. After that, we calculate the Value matrix as

$$V = F_L W_{VC} \tag{10}$$

where W_{VC} is a learnable weight matrix. Finally, the channel-wise GLocal feature F_C can be expressed as

$$F_C = VA_C. \tag{11}$$

Given both F_P and F_C, the final GLocal feature can be generated by combining them with an element-wise addition

$$F_G = F_P + F_C. \tag{12}$$

Additionally, we apply a residual connection between the LFA block and the GFL block

$$F_{\text{OUT}} = F_L + \text{LBR}(F_G) \tag{13}$$

where F_{OUT} is the final output feature map of the defined GFL block, and LBR denotes the combination of Linear, BatchNorm, and ReLU layers.

IV. EXPERIMENTS

In this section, we first introduce the implementation details of our 3DGTN, including hardware configuration and hyperparameter settings. Second, we present the performance evaluation of our method on classification and segmentation tasks, comparing it to state-of-the-art methods. Specifically, we tested our method for the classification task on the widely used ModelNet40 and ScanObjectNN datasets [22], [42]. For object part segmentation, we tested our method on the ShapeNet dataset [43]. For semantic segmentation, we tested our method on the challenging large-scale MS-LiDAR dataset [44]. Finally, we present the ablation experiment results on the main components of our method.

A. Implementation Details

We implemented our classification and segmentation networks in PyTorch. Both were trained and tested on an NVIDIA Tesla V100 GPU. We used the SGD Optimizer with a momentum of 0.9 and weight decay of 0.0001. The initial learning rate was set to 0.01, with a cosine annealing schedule. We trained classification, part segmentation, and semantic segmentation networks for 250, 300, and 500 epochs, respectively, with the same batch size of 16.

B. Point Cloud Classification

1) Datasets and Metrics: The ModelNet40 dataset contains 12 311 CAD models with 40 object categories. We split them into 9843 training samples and 2468 testing models, following PoineNet++ [23]. For a fair comparison, we downsampled each input point cloud to 1024 points with normals via FPS. Since point clouds in ModelNet40 are generated from 3-D meshes, we can easily obtain the normal of each point according to the corresponding surface normal. The mean accuracy within each category (mAcc) and the overall accuracy (OA) are used for performance evaluation, which are formulated as

$$\text{mAcc} = \frac{\sum_{i=1}^{K} \frac{T_i}{N_i}}{K}$$
$$\text{OA} = \frac{T}{N} \tag{14}$$

where T is the number of all correctly predicted point clouds, $T = \sum_{i=1}^{K} T_i$, T_i is the number of correctly predicted point clouds in class i, K is the number of classes in the dataset, and N is the number of all point clouds in the dataset, $N = \sum_{i=1}^{K} N_i$ and N_i is the number of point clouds in class i. Additionally, we adopt the total number of parameters, FLOating point operations (FLOPs), and Frame Per Second to evaluate the model size and efficiency.

To further evaluate the performance of 3DGTN to the real-world data captured by LiDAR scanning, ScanObjectNN [42] classification performance was also tested in our experiments. There are ∼15 000 objects in ScanObjectNN, which are categorized into 15 categories with 2902 unique object instances. Since each object was segmented from the scanned scene point cloud, object point clouds usually include numerous outliers in the form of background points, and were corrupted by occlusions and noises. Therefore, it was more challenging to perform shape classification on this dataset. We used the hardest variant of the dataset (PB_T50_RS) and adopted the original training/testing split as in [42]. Similarly, each sample from ScanObjectNN was downsampled to 1024 points. Since point clouds in PB_T50_RS have no normal information, we only took the 3-D coordinates of point clouds as input.

2) Performance Comparison: We compared our 3DGTN with the state-of-the-art Transformer-based methods and other deep learning-based methods. The comparison results are shown in Tables I and II. Specifically, for the ModelNet40 dataset, our method achieves the best mean accuracy of 92.4% among all benchmarked methods in terms of mAcc, outperforming the prior state-of-the-art (PointMLP [29]) by 1.0 absolute percentage points. In terms of OA, our method achieves the best result of 94.0% among the Transformer-based methods. For the model size, our method requires fewer parameters (5.21 MB) and FLOPs (3.09 GB) compared to most Transformer-based algorithms, accounting for only 57% and 18% of PT [11], respectively. However, due to the naive implementation of several time-consuming operations like downsampling and kNN neighborhood construction, the inference speed of our method can still be improved. For the ScanobjectNN dataset, 3DGTN also achieves competitive performance with the SOTA approaches. Especially, it obtains the best results in terms of both OA (85.8%) and mAcc (83.2%) among all compared Transformer-based methods, which demonstrates the excellent performance of 3DGTN in LiDAR data processing.

3) Visualization: We generate Grad-CAM [45] map visualization results from the ModelNet40 dataset. The Grad-CAM technique is designed to produce a coarse localization map highlighting the important regions in target point clouds. It uses the gradient information flowing into the last convolutional layer of a deep network to understand the importance of each neuron for a decision of interest. As shown in Fig. 5, we obtain the regions of interest of our network for several point clouds of the Airplane, Car, Cup, and Plant classes. From the results, the attention (colored in red) is mainly focused on the wings and tail of the Airplane, the tires of the Car, the handle of the Cup, and the leaves of the Plant. As we can

TABLE I

QUANTITATIVE COMPARISON (mACC, OA, PARAMETERS, FLOPs, AND FRAME PER SECOND) OF CLASSIFICATION PERFORMANCE ON THE MODELNET40 DATASET. TRANSFORMER METHODS ARE SEPARATED FROM OTHER LEARNING-BASED METHODS. THE HIGHEST EVALUATION SCORE IS SHOWN IN BOLD TYPE

Methods	Input Size	mAcc (%)	OA (%)	Parameters (MB)	FLOPs (GB)	Frame Per Sec.
Other Learning-based Methods						
3DShapeNets [22]	1024	77.3	84.7	-	-	-
PointNet [7]	1024	86.0	89.2	3.47	0.45	**614**
PointNet++ [23]	1024	88.2	91.9	1.74	4.09	16
diffConv [24]	1024	90.4	93.2	2.08	**0.16**	-
CurveNet [25]	1024	90.4	93.1	-	-	-
PointCNN [26]	1024	88.1	92.2	**0.6**	1.54	14
DGCNN [20]	1024	90.2	92.2	1.81	2.43	279
FatNet [27]	1024	90.6	93.2	-	-	-
DRNet [28]	1024	-	93.1	-	-	-
PointMLP [29]	1024	**91.4**	**94.1**	12.6	-	112
Point-PN [30]	1024	-	93.8	0.8	-	532
RepSurf [31]	1024	91.1	94.0	1.48	-	205
Transformer-based Methods						
PATs [32]	1024	-	91.7	-	-	-
LFT-Net [12]	1024	89.7	93.2	-	-	-
PointTransformer [33]	1024	89.0	92.8	13.86	9.36	17
MLMST [34]	1024	-	92.9	-	-	-
PointCloudTransformer [6]	1024	90.3	93.2	**2.80**	**2.02**	125
LSLPCT [5]	1024	90.5	93.5	-	-	-
PointTransformer [11]	1024	90.6	93.7	9.14	17.14	15
CloudTransformers [35]	1024	90.8	93.1	22.91	12.69	12
GBNet [21]	1024	91.0	93.8	8.38	9.02	102
3DCTN [14]	1024	91.6	93.2	4.21	3.76	24
PatchFormer [17]	1024	-	93.5	2.45	1.62	**201**
Ours	1024	**92.4**	**94.0**	5.21	3.09	15

TABLE II

QUANTITATIVE COMPARISON (%) OF CLASSIFICATION PERFORMANCE ON THE SCANOBJECTNN DATASET. TRANSFORMER-BASED METHODS AND OTHER LEARNING-BASED METHODS ARE SEPARATED. THE HIGHEST EVALUATION SCORE IS SHOWN IN BOLD TYPE

Methods	Input Size	mAcc (%)	OA (%)
Other Learning-based Methods			
PointNet [7]	1024	63.4	68.2
PointNet++ [23]	1024	75.4	77.9
SpiderCNN [36]	1024	69.8	73.7
PointCNN [26]	1024	75.1	78.5
DGCNN [20]	1024	73.6	78.1
SimpleView [37]	1024	-	80.5
PRANet [38]	1024	79.1	82.1
PointMLP [29]	1024	**84.4**	85.7
RepSurf [31]	1024	83.1	**86.0**
Transformer-based Methods			
PointTransformer [33]	1024	75.3	77.6
PointCloudTransformer [6]	1024	77.1	80.5
PointTransformer [11]	1024	78.2	80.8
GBNet [21]	1024	77.8	80.5
Point-MAE [39]	1024	-	85.2
Point-TnT [40]	1024	-	83.5
Point-BERT [41]	1024	-	83.1
3DCTN [14]	1024	79.5	81.5
Ours	1024	**83.2**	**85.8**

see, all the regions of interest are consistent with the human visual system, which helps us establish appropriate trust in predictions from deep networks.

C. Part Segmentation on ShapeNet Dataset

1) Dataset and Metrics: The ShapeNet dataset contains 16 880 models with 16 shape categories. We split them into 14 006 training samples and 2874 testing models, following PT [11]. The dataset has 50 part labels, and each object has at least two parts. For a fair comparison, we downsampled each input point cloud to 2048 points with normals by FPS. The category-wise mean intersection over union (mIoU) and instance-wise mIoU [11] are used for performance evaluation, which are formulated as below

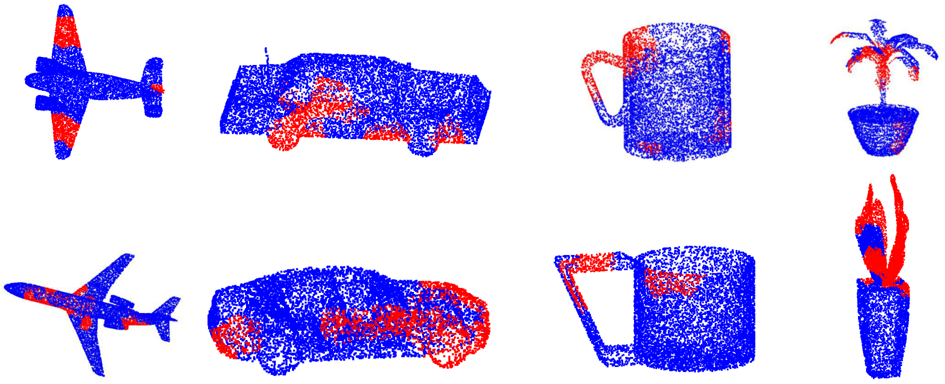
$$
\begin{aligned}
\text{Cat.mIoU} &= \frac{\sum_{i=1}^{\text{Cls}} \sum_{j=1}^{H_i} \text{mIoU}_j}{\text{Cls}} \\
\text{mIoU}_j &= \frac{\sum_{i=1}^{M_j} \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i+\text{FN}_i}}{M} \\
\text{Ins.mIoU} &= \frac{\sum_{i=1}^{G} \text{mIoU}_i}{G}
\end{aligned}
\tag{15}
$$

where Cls is the number of total shape classes of the dataset (Cls = 16 in the ShapeNet dataset), H_i represents the number of instances of the class i, M_j represents the number of part classes (varies with shape classes) in the jth instance, TP_i represents the number of the true positive samples in the ith part class, and G is the numbers of all instances in the dataset ($G = 2874$ in the testing dataset of ShapeNet.)

2) Performance Comparison: The comparison results are shown in Table III. As measured by instance-wise mIoU, our 3DGTN achieves competitive results (86.6%) compared with the SOTA Transformer-based methods such as Stratified Transformer [9]. This demonstrates the excellent performance of 3DGTN in terms of part segmentation. Several part segmentation results are shown in Fig. 6.

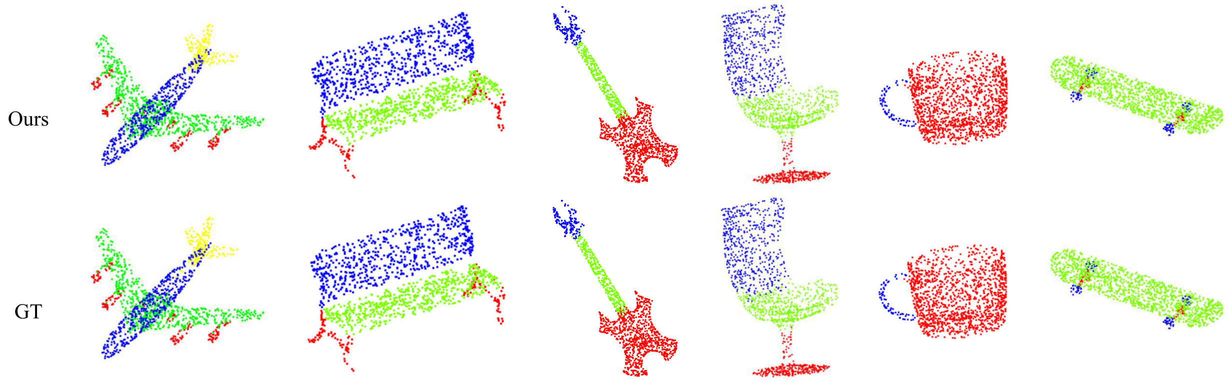
D. Semantic Segmentation on Airborne MS-LiDAR Dataset

1) Dataset and Metrics: Most recently, a large-scale airborne MS-LiDAR dataset was proposed in [44]. We tested 3DGTN on this dataset to explore its performance in practical remote sensing applications. The MS-LiDAR dataset was



(a) (b) (c) (d)

Fig. 5. Visualization of 3DGTN attention on the ModelNet40 classification dataset. As can be seen, the attention (red) is focused on the discriminative parts of targets, such as the wings of (a) airplane, the tires of (b) car, the handle of (c) cup, and the leaves of (d) plant.



(a) (b) (c) (d) (e) (f)

Fig. 6. Part segmentation results from the ShapeNet dataset. As can be seen, our segmentation predictions are faithful to the ground truth. (a) Airplane. (b) Chair 1. (c) Guitar. (d) Chair 2. (e) Mug. (f) Skateboard.

TABLE III

QUANTITATIVE COMPARISON (%) OF PART SEGMENTATION PERFORMANCE ON THE SHAPENET DATASET. THE HIGHEST EVALUATION SCORE IS SHOWN IN BOLD

Methods	Cat. mIoU	Ins. mIoU
PointNet [7]	80.4	83.7
PointNet++ [23]	81.9	85.1
A-SCN [46]	-	84.6
PCNN [47]	81.8	85.1
SpiderCNN [36]	82.4	85.3
SPLATNet [48]	83.7	85.4
SGPN [49]	82.8	85.8
SubSparseCNN [50]	83.3	86.0
PointCNN [26]	84.6	86.1
PointConv [51]	82.8	85.7
Point2Sequence [52]	-	85.2
DGCNN [20]	82.3	85.2
PVCNN [53]	-	86.2
RSCNN [54]	84.0	86.2
KPConv [55]	85.0	86.2
InterpCNN [56]	84.0	86.3
DensePoint [57]	84.2	86.4
PAConv [58]	84.6	86.1
PointTransformer [11]	83.7	**86.6**
StratifiedTransformer [9]	**85.1**	**86.6**
PatchFormer [17]	-	86.5
APES [59]	83.7	85.8
Ours	84.0	**86.6**

captured by a Teledyne Optech Titan MS-LiDAR system [44]. In addition to 3-D coordinates, each point also has three channels with wavelengths of 1550 nm (MIR), 1064 nm (NIR), and 532 nm (Green). The dataset was labeled into six categories: Road, Building, Grass, Tree, Soil, and Powerline. The dataset was divided into 13 subsets, where subsets 1–10 were taken as training data, while subsets 11–13 were taken as testing data. For fair comparison, we took the same data preprocessing (data fusion, normalization, and training/testing sample generation) methods described in [44]. The average F_1 score [62], mIoU, and OA are used for performance evaluation

$$\begin{aligned}\mathrm{mIoU} &= \frac{\sum_{i=1}^{\mathrm{Cls}} \mathrm{mIoU}_i}{\mathrm{Cls}}\\ \mathrm{mIoU}_i &= \frac{\sum \mathrm{TP}_i}{\sum \mathrm{TP}_i + \sum \mathrm{FP}_i + \sum \mathrm{FN}_i}\\ \mathrm{Average} F_1 &= \frac{\sum_{i=1}^{\mathrm{Cls}} F_{1i}}{\mathrm{Cls}}\end{aligned} \tag{16}$$

where Cls = 6 in the airborne MS-LiDAR dataset, and F_{1i} is calculated as follows:

$$\begin{aligned}F_{1i} &= \frac{\mathrm{Precision}_i * \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i}\\ \mathrm{Precision}_i &= \frac{\sum \mathrm{TP}_i}{\sum \mathrm{TP}_i + \sum \mathrm{FP}_i}\\ \mathrm{Recall}_i &= \frac{\sum \mathrm{TP}_i}{\sum \mathrm{TP}_i + \sum \mathrm{FN}_i}.\end{aligned} \tag{17}$$

TABLE IV

Confusion Matrix of 3DGTN on the Airborne MS-Lidar Dataset. The Second to Seventh Row Represent the Number of Points, the Last Three Rows Represent the Precision, Recall, and F_1 Score in % for Each Class

Categories	Road	Building	Grass	Tree	Soil	Powerline
Road	200355	14692	49	7	6274	0
Building	29709	871560	11144	612	14449	0
Grass	203	7869	925649	1281	18	424
Tree	71	648	2513	108529	2	14
Soil	6691	10628	103	162	26802	0
Powerline	5	15	323	0	0	7036
Precision	90.50	93.97	98.95	97.09	60.38	95.35
Recall	84.53	96.26	98.50	98.14	56.37	94.14
F_1	87.44	95.10	98.73	97.61	58.31	94.74

TABLE V

Quantitative Comparison (%) of Semantic Segmentation Performance on the Airborne MS-LiDAR Dataset. The Highest Evaluation Score Is Shown in Bold. The F_1 Score for Each Category Is Also Provided

Methods	Road	Building	Grass	Tree	Soil	Powerline	Average F_1	mIoU	OA
PointNet [7]	50.81	79.20	68.61	75.21	12.73	22.56	51.52	44.28	83.79
PointNet++ [23]	71.08	83.98	93.24	96.45	30.24	57.28	72.05	58.60	90.09
DGCNN [20]	70.42	90.25	93.62	97.93	21.97	55.24	71.57	51.04	91.36
RSCNN [54]	71.18	89.00	91.42	95.63	26.43	70.03	73.90	56.10	90.99
GACNet [60]	64.51	84.21	93.41	96.66	22.77	33.83	67.65	51.04	89.91
SE-PointNet++ [61]	70.32	85.64	94.70	97.05	37.02	70.35	75.84	60.15	91.16
FR-GCNet [44]	82.63	90.81	95.33	98.77	28.72	74.11	78.61	65.78	93.55
Xiao et al. [62]	73.33	90.51	86.30	95.20	**59.24**	95.60	83.30	79.25	94.04
GCNAS [63]	**87.75**	**98.68**	96.00	**99.49**	50.74	**96.12**	88.13	**82.23**	95.19
Ours	87.44	95.10	**98.73**	97.61	58.31	94.74	**88.63**	82.05	**95.20**

TABLE VI

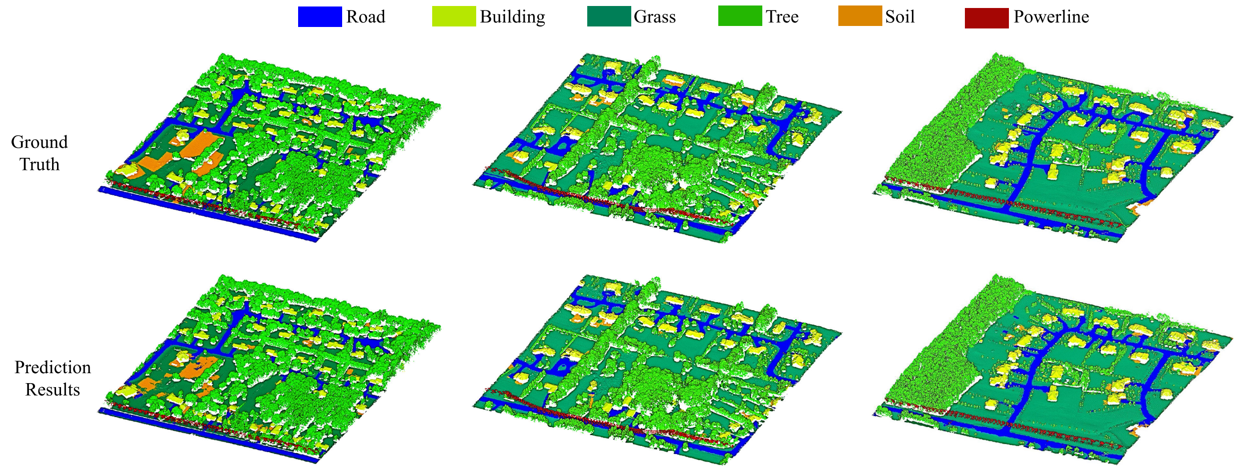
Results (%) Comparison of Input Data With Removal of Different Channel on the Airborne MS-LiDAR Dataset. The F_1 Score for Each Category Is Also Provided

Input				Road	Building	Grass	Tree	Soil	Powerline	Average F_1	mIoU	OA
XYZ	MIR	NIR	Green									
✓	-	✓	✓	84.06	90.70	95.78	96.19	51.22	91.12	84.85	79.92	93.90
✓	✓	-	✓	60.62	84.59	93.30	90.63	33.25	83.37	74.29	69.76	86.41
✓	✓	✓	-	83.08	90.83	92.66	96.95	45.19	88.05	82.79	78.73	92.09
✓	✓	✓	✓	**87.44**	**95.10**	**98.73**	**97.61**	**58.31**	**94.74**	**88.63**	**82.05**	**95.20**

Additionally, the F_{1_i} score for each category i is also provided.

2) Performance Comparison: As shown in Table IV, the semantic segmentation results of Airborne MS-LiDAR data are presented in the form of a confusion matrix. Since we integrate the CSA mechanism into global feature learning, our method is able to handle MS LiDAR point cloud segmentation well. Specifically, from the table, the number of samples differs significantly among categories. In this case of extremely imbalanced data, our 3DGTN still achieves excellent F_1 scores of over 85% for all categories except soil. The F_1 scores of the grass, tree, and building are over 95%. However, since the geometric characteristics of the soil are very similar to those of grass, which tends to confuse the network, the segmentation results of the soil are not very satisfactory. More feature discrimination approaches would be designed in our future work which could improve the segmentation of similar classes. The comparison results are shown in Table V. As can be seen, our 3DGTN outperforms all benchmarked methods in terms of average F_1 score (88.63%). It surpasses the prior SOTA methods such as [62] and [63] by 5.33 and 0.50 absolute percentage points, respectively. It also achieves the best OA (95.20%) and a competitive mIoU (82.05%). The prediction results and corresponding ground truth of testing data are shown in Fig. 7. These results demonstrate that our method has excellent performance in processing real-scanned data, exceeding previous SOTA.

We also explored the importance of different channels in the MS-LiDAR data. Specifically, we removed the each of the three channels (MIR, NIR, and Green) of MS-LiDAR data, and then analyzed the corresponding performance changes in Table VI. When the NIR channel was removed (Row 4), the performance dropped significantly (average F_1 score was reduced to 74.29% from 88.63%). There is also a slight performance drop when the MIR or Green channel is removed. The results demonstrate that all these three channels are useful for data segmentation, where the NIR channel contributes the most to performance.



(a) (b) (c)

Fig. 7. Semantic segmentation results and ground truth from the Airborne MS-LiDAR dataset. (a) Test area-11. (b) Test area-12. (c) Test area-13.

TABLE VII

Quantitative Comparison Results of Ablation Studies for the Main Components of 3DGTN, Which Were Performed on the ModelNet40 Classification Dataset. − Means Component Removal, and → Means Component Changing

Ablation		mAcc (%)	OA (%)	Parameters (MB)	FLOPs (GB)	Frame Per Sec.
Local feature aggregation	Graph convolution → Standard MLP	91.6	93.1	5.18	0.45	17
	Multi-scale → Single-scale	91.3	92.9	2.11	0.64	19
	FPS → Random sampling	92.1	93.5	5.21	3.09	23
GLocal feature learning	−	91.2	92.7	2.25	2.78	17
	− CSA	91.9	93.7	4.71	3.04	16
	− PPSA	91.3	93.2	3.75	2.95	16
	PPSA → Vanilla PSA	91.5	93.6	5.11	3.08	15
Multi-level GLocal feature	−	92.1	93.6	4.75	3.09	15
3DGTN		**92.4**	**94.0**	5.21	3.09	15

E. Ablation Study

We conducted a series of ablation experiments for the main components of our 3DGTN to verify their effectiveness. These experiments were performed on the ModelNet40 dataset.

1) LFA Block: We first investigate the effectiveness of the LFA block, which is used to capture local information. As shown in Table VII Row 2, the performance with the MLP-based LFA block is 91.6%/93.1% in terms of mAcc/OA, which is lower than that with the initial LFA block (92.4%/94.0%). This demonstrates that the GCN-based LFA block plays an important role in our algorithm. We also replaced the multiscale strategy of the LFA block with the single-scale one. As shown in Table VII Row 3, the classification performance of the multiscale strategy is superior (91.3%/92.9%). This suggests that the multiscale features are beneficial to enhancing the expression of local information, thereby improving the performance of our algorithm. Finally, we replaced the Furthest Point Sampling method with random sampling, to investigate the performance of 3DGTN with different sampling approaches. As shown in Table VII Row 4, the classification accuracy drops slightly with random sampling (92.1%/93.5% in terms of mAcc/OA). This is because compared with random sampling, FPS could maintain the geometric characteristics of the target point cloud better. However, as measured by Frame Per Second, Furthest Point Sampling (15) is more time-consuming than random sampling (23). Therefore, developing an efficient and adaptive sampling method for point cloud processing is one of our future works.

2) GFL Block: We conducted a detailed ablation study on the GFL block. As shown in Table VII, when we removed the GFL block, the performance drops significantly, which demonstrates that the GFL block is essential to our algorithm. Second, since the GFL block contains two important mechanisms: PPSA and CSA, we also studied the effectiveness of each mechanism. When the CSA was removed, the classification accuracy (mAcc/OA) drops from 92.4%/94.0% to 91.9%/93.7%. Likewise, when the PPSA was removed, there is a similar drop (from 92.4%/94.0% to 91.13%/93.2%). These results suggest that both self-attention mechanisms are effective in improving classification performance. Additionally, to further verify the effectiveness of the PPSA mechanism, we replaced it with a regular point-wise self-attention mechanism (treating the F_L as the Value matrix). After replacing, we observe a 0.9% and 0.4% drop in mAcc and OA, respectively. This confirms the superiority of our PPSA mechanism.

3) Multilevel GLocal Feature Concatenation: We studied the effectiveness of the multilevel GLocal feature concatenation. As illustrated in Fig. 1, we concatenate the output feature of each level (module) using a residual connection to generate the multilevel GLocal feature. As shown in Table VII Row 9, when the residual connection was removed, we observed a 0.3% and 0.4% drop in mAcc and OA, respectively. This suggests that the multilevel GLocal feature contributes significantly to performance improvement.

V. CONCLUSION

In this article, we have proposed a hierarchical point cloud representation network for classification and segmentation, named 3DGTN. It is an encoder–decoder architecture. The encoder has a series of GLocal modules for effective feature extraction, each of which consists of two cascaded LFA and GFL blocks. In particular, for the GFL block, we adopt the dual-attention Transformer which combines the PPSA and CSA mechanisms. The novel PPSA mechanism is designed to fuse both global features and local neighborhood information of input points, which is able to improve feature learning ability as GLocal features and mitigate local information loss. The decoder is composed of several MLP layers for efficient point cloud reconstruction. It achieves a better trade-off between accuracy and efficiency than a symmetric decoder. Extensive experiments on the ModelNet40, ScanObjectNN classification datasets [22], [42], ShapeNet part segmentation dataset [43], and MS-LiDAR semantic segmentation dataset [44] demonstrate the superiority of our method in dealing with both synthetic data and real-scene LiDAR data.

Future Work: Our hierarchical network uses Euclidean distance-based downsampling and neighborhood search methods, which are time-consuming and cannot serve the semantic information extracted by the network very well. Since the attention map in the Transformer contains rich feature relationships, we plan to utilize the attention map for semantic-based point cloud sampling and grouping as a future research project. To this end, the "superpoint" strategy could be a potential solution.

REFERENCES

[1] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, 2017, pp. 6000–6010.

[2] D. Lu, Q. Xie, M. Wei, K. Gao, L. Xu, and J. Li, "Transformers in 3D point clouds: A survey," 2022, *arXiv:2205.07417*.

[3] X. Qiang, W. He, S. Chen, Q. Lv, and F. Huang, "Hierarchical point cloud transformer: A unified vegetation semantic segmentation model for multisource point clouds based on deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, doi: 10.1109/TGRS.2023.3336651.

[4] Z. Luo, Z. Zeng, W. Tang, J. Wan, Z. Xie, and Y. Xu, "Dense dual-branch cross attention network for semantic segmentation of large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, doi: 10.1109/TGRS.2023.3341894.

[5] Y. Song, F. He, Y. Duan, T. Si, and J. Bai, "LSLPCT: An enhanced local semantic learning transformer for 3-D point cloud analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4708813, doi: 10.1109/TGRS.2022.3202823.

[6] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, Jun. 2021.

[7] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[8] X.-F. Han, Z.-Y. He, J. Chen, and G.-Q. Xiao, "3CROSSNet: Cross-level cross-scale cross-attention network for point cloud representation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3718–3725, Apr. 2022.

[9] X. Lai et al., "Stratified transformer for 3D point cloud segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8500–8509.

[10] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[11] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[12] Y. Gao, X. Liu, J. Li, Z. Fang, X. Jiang, and K. M. S. Huq, "LFT-Net: Local feature transformer network for point clouds analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2158–2168, Feb. 2023, doi: 10.1109/TITS.2022.3140355.

[13] Z. Liu, X. Yang, H. Tang, S. Yang, and S. Han, "FlatFormer: Flattened window attention for efficient point cloud transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1200–1211.

[14] D. Lu, Q. Xie, K. Gao, L. Xu, and J. Li, "3DCTN: 3D convolution-transformer network for point cloud classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24854–24865, Dec. 2022, doi: 10.1109/TITS.2022.3198836.

[15] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107446, doi: 10.1016/j.patcog.2020.107446.

[16] L. Hui, H. Yang, M. Cheng, J. Xie, and J. Yang, "Pyramid point cloud transformer for large-scale place recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6098–6107.

[17] C. Zhang, H. Wan, X. Shen, and Z. Wu, "PatchFormer: An efficient point transformer with patch attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11789–11798.

[18] J. Sun, C. Qing, J. Tan, and X. Xu, "Superpoint transformer for 3D scene instance segmentation," in *Proc. Conf. AAAI Artif. Intell.*, vol. 37, Jun. 2023, pp. 2393–2401.

[19] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D semantic segmentation with superpoint transformer," 2023, *arXiv:2306.08045*.

[20] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[21] S. Qiu, S. Anwar, and N. Barnes, "Geometric back-projection network for point cloud classification," *IEEE Trans. Multimedia*, vol. 24, pp. 1943–1955, 2022.

[22] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[23] 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. Process. Syst.*, 2017, pp. 5105–5114.

[24] M. Lin and A. Feragen, "DiffConv: Analyzing irregular point clouds with an irregular view," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 380–397.

[25] A. A. M. Muzahid, W. Wan, F. Sohel, L. Wu, and L. Hou, "CurveNet: Curvature-based multitask learning deep networks for 3D object recognition," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1177–1187, Jun. 2021, doi: 10.1109/JAS.2020.1003324.

[26] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 820–830.

[27] C. Kaul, N. Pears, and S. Manandhar, "FatNet: A feature-attentive network for 3D point cloud processing," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 7211–7218, doi: 10.1109/ICPR48806.2021.9412731.

[28] S. Qiu, S. Anwar, and N. Barnes, "Dense-resolution network for point cloud classification and segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3813–3822, doi: 10.1109/WACV48630.2021.00386.

[29] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=3Pbra-_u76D

[30] R. Zhang, L. Wang, Y. Wang, P. Gao, H. Li, and J. Shi, "Starting from non-parametric networks for 3D point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5344–5353.

[31] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 18942–18952.

[32] J. Yang et al., "Modeling point clouds with self-attention and Gumbel subset sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3323–3332, doi: 10.1109/CVPR.2019.00344.

[33] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134826–134840, 2021, doi: 10.1109/ACCESS.2021.3116304.

[34] Q. Zhong and X.-F. Han, "Point cloud learning with transformer," 2021, *arXiv:2104.13636*.

[35] K. Mazur and V. Lempitsky, "Cloud transformers: A universal approach to point cloud processing tasks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10695–10704, doi: 10.1109/ICCV48922.2021.01054.

[36] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11212, 2018, pp. 90–105.

[37] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng, "Revisiting point cloud shape classification with a simple and effective baseline," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 3809–3820.

[38] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "PRA-Net: Point relation-aware network for 3D point cloud analysis," *IEEE Trans. Image Process.*, vol. 30, pp. 4436–4448, 2021.

[39] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 604–621.

[40] A. Berg, M. Oskarsson, and M. O'Connor, "Points to patches: Enabling the use of self-attention for 3D shape recognition," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 528–534.

[41] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-training 3D point cloud transformers with masked point modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19313–19322.

[42] M. A. Uy, Q. Pham, B. Hua, T. Nguyen, and S. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1588–1597.

[43] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, 2016.

[44] P. Zhao et al., "Airborne multispectral LiDAR point cloud classification with a feature reasoning-based graph convolution network," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 105, Dec. 2021, Art. no. 102634.

[45] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[46] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for point cloud recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4606–4615.

[47] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," 2018, *arXiv:1803.10091*.

[48] H. Su et al., "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539, doi: 10.1109/CVPR.2018.00268.

[49] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2569–2578.

[50] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[51] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630, doi: 10.1109/CVPR.2019.00985.

[52] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, "Point2sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8778–8785.

[53] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. NIPS*, Vancouver, BC, Canada, 2019, pp. 963–973.

[54] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[55] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[56] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3D point cloud understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1578–1587.

[57] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5239–5248.

[58] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3173–3182.

[59] C. Wu, J. Zheng, J. Pfrommer, and J. Beyerer, "Attention-based point cloud edge sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 5333–5343.

[60] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10296–10305.

[61] Z. Jing et al., "Multispectral LiDAR point cloud classification using SE-PointNet++," *Remote Sens.*, vol. 13, no. 13, p. 2516, Jun. 2021.

[62] K. Xiao, J. Qian, T. Li, and Y. Peng, "Multispectral LiDAR point cloud segmentation for land cover leveraging semantic fusion in deep learning network," *Remote Sens.*, vol. 15, no. 1, p. 243, Dec. 2022.

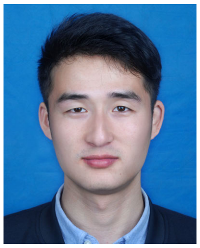
[63] Q. Zhang, Y. Peng, Z. Zhang, and T. Li, "Semantic segmentation of spectral LiDAR point clouds based on neural architecture search," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, doi: 10.1109/TGRS.2023.3284995.

Dening Lu received the B.Sc. and M.Sc. degrees in electrical engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in systems design engineering with the Geospatial Sensing and Data Intelligence Group, University of Waterloo, Waterloo, ON, Canada.

He has published papers in IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and ICCV. His research interests include 3-D point cloud processing and deep learning.

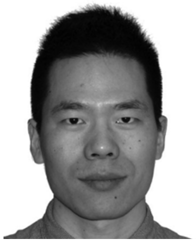
Kyle (Yilin) Gao (Graduate Student Member, IEEE) received the bachelor's degree in mathematics from the University of Waterloo, Waterloo, ON, Canada, in 2016, and the master's degree in physics from the University of Victoria, Victoria, BC, Canada, in 2020. He is currently pursuing the Ph.D. degree in systems design engineering with the Geospatial Sensing and Data Intelligence Group, University of Waterloo.

He has published papers in the *International Journal of Applied Earth Observation and Geoinformation, Geomatica*, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. His research interests include computer vision and deep learning.



Qian Xie received the B.Sc. and Ph.D. degrees in electrical engineering from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2014 and 2021, respectively.

He is currently a Research Associate with the Department of Computer Science, University of Oxford, Oxford, U.K. He works with the Cyber-Physical Systems Group under the supervision of Prof. Niki Trigoni and Prof. Andrew Markham. Prior to Oxford, he went to Cardiff University, Cardiff, U.K., as a joint-trained Ph.D. Student in 2019 for 18 months. He publishes extensively in venues and journals, such as CVPR, ICCV, ECCV, RA-L, TMM, and IJCV. His research interests include 3-D vision, point cloud processing, deep learning, and scene understanding.



Linlin Xu (Member, IEEE) received the B.Eng. and M.Sc. degrees in geomatics engineering from China University of Geosciences (Beijing), Beijing, China, in 2007 and 2010, respectively, and the Ph.D. degree in geography from the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada, in 2014.

He is currently a Research Assistant Professor with the Department of Systems Design Engineering, University of Waterloo. He has published various papers on high-impact remote sensing journals and conferences. His research interests include hyperspectral imaging, synthetic aperture radar and LiDAR, machine learning, and environmental monitoring.

Jonathan Li (Fellow, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa, in 2000.

He is currently a Professor of geomatics and systems design engineering, University of Waterloo, Waterloo, ON, Canada. He has supervised more than 120 master's and Ph.D. Students as well as Post-Doctoral Fellows to completion. He has coauthored more than 490 publications, more than 320 of which were published in refereed journals, including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), ISPRS-JPRS, RSE, and JAG. He has also published papers in flagship conferences in computer vision and AI, including CVPR, AAAI, and IJCAI. His main research interests include AI-based information extraction from Earth observation images and LiDAR point clouds, and 3-D vision and GeoAI.

Dr. Li is a fellow of the Royal Society of Canada (RSC) Academy of Science, the Canadian Academy of Engineering (CAE), and the Engineering Institute of Canada (EIC). He is the Editor-in-Chief of *International Journal of Applied Earth Observation and Geoinformation* (JAG) and an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.