Transformer-based Point Cloud Processing and Analysis for LiDAR Remote Sensing

by

Dening Lu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2025

© Dening Lu 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. Derek D. Lichti, Professor Dept. of Geomatics Engineering, University of Calgary
Supervisor(s):	Dr. Jonathan Li, Professor Dept. of Geography & Environmental Management, University of Waterloo
Supervisor(s):	Dr. Linlin Xu, Assistant Professor Dept. of Geomatics Engineering, University of Calgary
Internal Member:	Dr. John Zelek, Associate Professor Dept. of Systems Design Engineering, University of Waterloo
Internal Member:	Dr. Nasser Lashgarian Azad, Professor Dept. of Systems Design Engineering, University of Waterloo
Internal-External Member:	Dr. Michael A. Chapman, Professor Dept. of Civil Engineering (Adjunct GEM), Toronto Metropolitan University

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This doctoral thesis is accomplished in the manuscript option, following the graduation guidelines administered by the Graduate Program in Systems Design Engineering. It consists of three required manuscripts, which form the main body of Chapters 3, 4 and 5 of this thesis. These manuscripts contain material from three multi-authored papers, where the manuscripts are minorly changed for formatting consistency. As the lead author of these three papers, I was responsible for conceptualization, methodology, code writing, validation experiments, investigation, data curation, writing, and visualization. My co-authors (Jun Zhou, Kyle Gao and Qian Xie) contributed to the manuscript review and editing and part of the comparison experiments. My supervisors, Prof. Jonathan Li and Prof. Linlin Xu, provided guidance throughout the process and feedback on draft manuscripts. The references for the three papers are provided below:

(1) **Dening Lu**, Kyle (Yilin) Gao, Qian Xie, Linlin Xu and Jonathan Li. 3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1-13, 2024.

(2) **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. Efficient Point Transformer with Dynamic Token Aggregating. *Expert Systems with Applications*, under review, submitted at Mar 17, 2025, arXiv preprint arXiv:2405.15827.

(3) **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. 3DLST: 3D Learnable Supertoken Transformer for LiDAR Point Cloud Scene Segmentation. *International Journal of Applied Earth Observation and Geoinformation*, under revision, submitted at January 06, 2025, arXiv preprint arXiv:2405.15826.

In addition to the three key manuscripts highlighted, my other publications have also provided a solid foundation and valuable experimental experience for this research topic. These works have not only contributed to the development of essential methodologies and insights but have also served as important references for understanding the background and related work, enriching the contextual framework of this thesis. The references are provided below:

(1) **Dening Lu**, Qian Xie, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. 3DCTN: 3D Convolution-Transformer Network for Point Cloud Classification. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24854-24865, 2022.

(2) **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Jing Du, Linlin Xu and Jonathan Li. Dynamic clustering transformer network for point cloud segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 128:103791, 2024.

(3) **Dening Lu**, Qian Xie, Mingqiang Wei, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. Transformers in 3d point clouds: A survey. arXiv preprint arXiv:2205.07417, 2022.

I understand that my thesis may be made electronically available to the public.

Abstract

The processing and analysis of Light Detection and Ranging (LiDAR) point cloud data, a fundamental task in Three-Dimensional (3D) computer vision, is essential for a wide range of remote sensing applications. However, the disorder, sparsity, and uneven spatial distribution of LiDAR point clouds pose significant challenges to effective and efficient processing. In recent years, Transformers have demonstrated notable advantages over traditional deep learning methods in computer vision, yet designing Transformer-based frameworks tailored to point clouds remains an underexplored topic. This thesis investigates the potential of Transformer models for accurate and efficient LiDAR point cloud processing.

Firstly, a 3D Global-Local (GLocal) Transformer Network (3DGTN) is introduced to capture both local and global context, thereby enhancing model accuracy for LiDAR data. This design not only ensures a comprehensive understanding of point cloud characteristics but also establishes a foundation for subsequent efficient Transformer frameworks.

Secondly, a fast point Transformer network with Dynamic Token Aggregation (DTA-Former) is proposed to improve model speed. By optimizing point sampling, grouping, and reconstruction, DTA-Former substantially reduces the time complexity of 3DGTN while retaining its strong accuracy.

Finally, to further reduce time and space complexity, a 3D Learnable Supertoken Transformer (3DLST) is presented. Building on DTA-Former, 3DLST employs a novel supertoken clustering strategy that lowers computational overhead and memory consumption, achieving state-of-the-art performance across multi-source LiDAR point cloud tasks in terms of both accuracy and efficiency.

These Transformer-based frameworks contribute to more robust and scalable LiDAR point cloud processing solutions, supporting diverse remote sensing applications such as urban planning, environmental monitoring, and autonomous navigation. By enabling efficient yet high-accuracy analysis of large-scale 3D data, this work fosters further research and innovation in LiDAR remote sensing.

Acknowledgements

Pursuing a Ph.D. has been a journey filled with challenges, growth, and invaluable experiences. As I reach this significant milestone, I would like to express my deepest gratitude to those who have supported and guided me along the way.

First and foremost, I extend my sincere appreciation to my supervisors, Professor Jonathan Li and Professor Lincoln Xu. Their unwavering support, insightful guidance, and encouragement have been instrumental in shaping my academic and research endeavors. Their mentorship has not only deepened my understanding of geospatial intelligence and remote sensing but has also fostered my ability to think critically and independently as a researcher.

I would also like to thank my thesis committee members: Professor Derek D. Lichti, Professor Nasser Lashgarian Azad, Professor Michael A. Chapman, and Professor John Zelek. Their valuable feedback, constructive criticism, and thoughtful discussions have greatly contributed to the refinement of my research. I am truly grateful for their time, expertise, and dedication to my academic progress.

I would like to express my gratitude to my colleagues and collaborators from the Geospatial Intelligence and Mapping (GIM) Lab. In particular, I am grateful to Kyle Gao, Dr. Hongjie He, Lanying Wang, Dr. Dedong Zhang, Weikai Tan, and Zhengsen Xu for their collaboration and shared enthusiasm for research. I also extend my sincere thanks to Dr. Liangzhi Li from Chang'an University, PhD student Jun Zhou from The Hong Kong Polytechnic University, and Dr. Qian Xie from the University of Leeds for their invaluable input and support in my research. Beyond research, I am deeply appreciative of the warmth and kindness I received from my lab members, William Xu and Hongzhang Xu, who have provided much-needed support in daily life. Your friendship and assistance have made my time in the lab more fulfilling and memorable.

A special thank you goes to Jing Du, whose incredible support during the final stages of my Ph.D. made a significant difference. Your help allowed me to dedicate more time and energy to my thesis writing and defense preparation, and I am truly grateful for your generosity and encouragement during this critical period.

I am also profoundly thankful to my NRC IRAP project team, particularly Dr. Shimon Schwartz and Dr. Norman (Norm) Vinson. This dynamic and supportive team has accompanied me throughout my doctoral journey, providing invaluable insights and assistance that have greatly enriched my research. I have been fortunate to work with such an inspiring group of professionals who have continuously motivated me.

Finally, my deepest gratitude goes to my parents, whose unconditional love and support have been the foundation of my achievements. Their belief in me has been a constant source of strength, and I am forever indebted to them for their patience, encouragement, and unwavering faith in my abilities.

This Ph.D. journey has been made all the more meaningful by the support of these incredible individuals, and I sincerely appreciate each and every one of them.

Table of Contents

Examining Committee ii

Author's Declaration iii

Statement of Contributions iv

Abstract vi

Acknowledgements vii

List of Figures xiii

List of Tables xv

List of Abbreviations xvii

1 Introduction 1

1.1 Background and Motivation 1

1.1.1 Literature Review 1

1.1.2 3D Transformers 3

1.2 Research Objectives 7

1.3 Thesis Structure 8

2 3D Transformers: An Overview **10**
2.1 Transformers in 3D Point Cloud Processing . . . 10
2.1.1 3D Point Cloud Classification . . . 10
2.1.2 3D Point Cloud Segmentation . . . 12
2.2 Efficient Point Transformers . . . 13
2.3 3D Self-attention Mechanism . . . 13
2.3.1 Point-wise Self-attention . . . 14
2.3.2 Channel-wise Self-attention . . . 17
2.4 Chapter Summary . . . 18

3 3DGTN: 3D GLocal Transformer Network **19**
3.1 Introduction . . . 19
3.2 Methodology . . . 20
3.2.1 Overview . . . 21
3.2.2 Local Feature Aggregation (LFA) Block . . . 21
3.2.3 Global Feature Learning (GFL) Block . . . 23
3.3 Experimental Results and Discussion . . . 25
3.3.1 Implementation Details . . . 25
3.3.2 Point Cloud Classification . . . 25
3.3.3 Part Segmentation on ShapeNet Dataset . . . 28
3.3.4 Semantic Segmentation on Airborne MS-LiDAR Dataset . . . 29
3.3.5 Ablation Study . . . 32
3.4 Chapter Summary . . . 33

4 Fast Point Transformer with Dynamic Token Aggregation **34**
4.1 Introduction . . . 34
4.2 Methodology . . . 36
4.2.1 Overview . . . 36

4.2.2 Learnable Token Sparsification (LTS) . . . 37
4.2.3 Dynamic Token Aggregation (DTA) . . . 39
4.2.4 Iterative Token Reconstruction (ITR) . . . 41
4.3 Experimental Results and Discussion . . . 42
4.3.1 Implementation Details . . . 42
4.3.2 Point Cloud Classification . . . 42
4.3.3 Point Cloud Part Segmentation . . . 43
4.3.4 Airborne MS-LiDAR Segmentation . . . 44
4.3.5 Ablation Study . . . 49
4.4 Chapter Summary . . . 50

5 3D Learnable Supertoken Transformer 52
5.1 Introduction . . . 52
5.2 Methodology . . . 53
5.2.1 Overview . . . 53
5.2.2 Dynamic Supertoken Optimization (DSO) . . . 54
5.2.3 Cross-Attention-guided Upsampling (CAU) . . . 56
5.3 Experimental Results and Discussion . . . 57
5.3.1 Implementation Details . . . 57
5.3.2 Airborne MS-LiDAR Dataset . . . 58
5.3.3 DALES Dataset . . . 61
5.3.4 Ablation Study . . . 63
5.4 Discussion . . . 65
5.5 Chapter Summary . . . 67

6 Conclusions and Recommendations 68
6.1 Conclusions . . . 68
6.2 Recommendations for Future Research . . . 69

References **70**

APPENDICES **88**

A Datasets and Metrics **89**

A.1 Results on the ScanObjectNN Dataset 89

A.2 Metrics Calculation 90

B Publications during the PhD Study **92**

C Copyright Acknowledgment for IEEE Materials **94**

List of Figures

1.1 Illustration of the Transformer encoder architecture. 4
1.2 Structure of the thesis. 8

2.1 Applications of Transformers in 3D point clouds. 11
2.2 Architectures of Point-wise self-attention variants. 15
2.3 Architectures of Channel-wise self-attention variants. 17

3.1 3DGTN networks for point cloud classification and segmentation. 20
3.2 Multi-scale Local Feature Aggregation (LFA) block. 22
3.3 Point-wise Self-Attention mechanism. 23
3.4 Channel-wise Self-Attention mechanism. 24
3.5 Visualization of 3DGTN attention on the ModelNet40 classification dataset. © 2022, IEEE 27
3.6 Part segmentation results of 3DGTN from the ShapeNet dataset. © 2022, IEEE 29
3.7 Semantic segmentation results of 3DGTN on the airborne MS-LiDAR dataset. 32

4.1 Pipelines of DTA-Former for classification (top) and segmentation (bottom). 37
4.2 Illustration of LTS results on Module 1. 38
4.3 Visualization of sparsified tokens from LTS blocks. 39
4.4 Illustration of the DTA block on Module 1. 40

4.5 Brief illustration of the segmentation network. 41
4.6 Illustration of part segmentation and WCA-map results on ShapeNet. . . 45
4.7 Comparison results from different methods on the testing area-11 in the airborne MS-LiDAR dataset. 46
4.8 Comparison results from different methods on the testing area-12 in the airborne MS-LiDAR dataset. 47
4.9 Comparison results from different methods on the testing area-13 in the airborne MS-LiDAR dataset. 48
4.10 Visualization results of samples from the testing area-11. 48

5.1 Architecture of 3DLST. 54
5.2 Illustration of the cross-attention calculation. 55
5.3 Illustration of the CAU block. 56
5.4 The training and validation loss curves during network training. 57
5.5 Comparison results from different methods on the testing area-11 in the airborne MS-LiDAR dataset. 59
5.6 Comparison results from different methods on the testing area-12 in the airborne MS-LiDAR dataset. 60
5.7 Comparison results from different methods on the testing area-13 in the airborne MS-LiDAR dataset. 60
5.8 Visualization results of supertoken clustering and prediction on the testing area-11 blocks. 61
5.9 Comparison results from different methods on the DALES dataset. 63
5.10 Memory Comparison of the proposed methods on the MS-LiDAR dataset. 65
5.11 Latency comparison of the proposed methods on the MS-LiDAR dataset. 66
5.12 Comparison results of different methods on the MS-LiDAR dataset. . . . 67

List of Tables

3.1 Quantitative comparison of 3DGTN on the ModelNet40 dataset. 26
3.2 Part segmentation results of 3DGTN on the ShapeNet dataset. 28
3.3 Comparison of the point numbers in each category in the MS-LiDAR dataset. 30
3.4 Confusion matrix (%) of 3DGTN on the airborne MS-LiDAR dataset. . . . 30
3.5 Quantitative comparison of 3DGTN on the airborne MS-LiDAR dataset. . 31
3.6 Results (%) comparison of input data with the removal of different channels on the airborne MS-LiDAR dataset. 31
3.7 Ablation studies for 3DGTN on the ModelNet40 dataset. 33

4.1 Quantitative comparison of DTA-Former on the ModelNet40 dataset. . . . 43
4.2 Part segmentation results of DTA-Former on the ShapeNet dataset. . . . 44
4.3 Confusion matrix (%) of DTA-Former-L on the airborne MS-LiDAR dataset. 46
4.4 Quantitative comparison of DTA-Former on the MS-LiDAR dataset. . . . 47
4.5 Ablation results for DTA-Former-L on the airborne MS-LiDAR dataset. . 49

5.1 Confusion matrix (%) of 3DLST on the airborne MS-LiDAR dataset. . . . 58
5.2 Quantitative comparison of 3DLST on the airborne MS-LiDAR dataset. . 59
5.3 Quantitative segmentation comparison (%) on the DALES dataset. 62
5.4 Detailed comparison between SPT [99] and 3DLST on the DALES dataset. 63
5.5 Results (%) of ablation study on the airborne MS-Lidar dataset. 64

5.6 Parameter sensitivity (%) of 3DLST to the number S of learnable supertokens on the airborne MS-LiDAR dataset. 64

A.1 Quantitative classification comparison (%) on the ScanObjectNN dataset. 90

A.2 Definition of TP, TN, FP, FN based on prediction and actual labels. . . . 91

List of Abbreviations

3D	Three-dimensional
3DGTN	3D Dual-attention GLocal Transformer Network
3DLST	3D Learnable Supertoken Transformer
A-SCN	Attentional ShapeContextNet
ALS	Aerial Laser Scanners
CAA	Channel-wise Affinity Attention
CAD	Computer-Aided Design
CAE	Channel Affinity Estimator
CAM	Cross-Attention Map
CAU	Cross-Attention-guided Upsampling
CCC	Compact Channel-wise Comparator
CLCA	Cross-Level Cross-Attention
CNN	Convolutional Neural Network
CSA	Channel-wise Self-Attention
CSCA	Cross-Scale Cross-Attention
CV	Computer Vision
DALES	Dayton Annotated Laser Earth Scan
DL	Deep Learning
DSO	Dynamic Supertoken Optimization
DTA	Dynamic Token Aggregation
DTA-Former	TransFormer with Dynamic Token Aggregation
DTNet	Dual Transformer Network
FPS	Farthest Point Sampling
GBNet	Geometric Backprojection Network
GCN	Graph Convolution Network
GFL	Global Feature Learning
GLocal	Global-Local

GSA	Group Shuffle Attention
GTNet	Graph Transformer Network
ITR	Iterative Token Reconstruction
KNN	K-Nearest Neighborhood
LFA	Local Feature Aggregation
LFT-Net	Local Feature Transformer Network
LTS	Learnable Token Sparsification
LiDAR	Light Detection And Ranging
MA	Max and Average
MLP	Multi-Layer Perceptron
MS	MultiSpectral
NLP	Natural Language Processing
OA	Overall Accuracy
P-A	Point Attention
PCT	Point Cloud Transformer
PSA	Point-wise Self-Attention
PT	Point Transformer
PVCNN	Point-Voxel CNN
RAM	Relation Attention Module
RELU	Rectified Linear Unit
RGB-D	Red Green Blue and Depth
SC-MSA	Shifted Channel Multi-head Self-Attention
SGD	Stochastic Gradient Descent
SOTA	State-Of-The-Art
SPT	SuperPoint Transformer
WCA	Weighted Cross-Attention
mIoU	Mean Intersection over Union

Chapter 1

Introduction

This chapter first introduces the research background and motivation, by reviewing the existing deep learning-based point cloud processing algorithms. Secondly, a detailed introduction to the 3D Transformer structure is presented, showing its advantages and highlighting the current research gaps. Thirdly, I summarize the research objectives of this thesis. Finally, the structure framework of this thesis is presented.

1.1 Background and Motivation

1.1.1 Literature Review

With the rapid advancement and increasing adoption of LiDAR sensors, including laser scanners and RGB-D cameras, the field of LiDAR point cloud processing has garnered significant attention. These sensors enable dense and direct sampling of object surfaces, producing massive and irregularly distributed point clouds that include spatial coordinates (X, Y, Z) and additional attributes such as reflection intensity. The simple and flexible data structure of point clouds makes them an effective medium for representing and analyzing the intricate three-dimensional structures of real-world environments.

In the field of point cloud processing and analysis, point cloud classification and segmentation are widely studied as mainstream and fundamental tasks. They play a critical role in various remote sensing applications such as urban planning [110, 37], environmental monitoring [33, 55], disaster management [102, 125], and more [12]. Remote sensing via LiDAR has emerged as a cornerstone technology for capturing large-scale, high-resolution

3D data. By enabling the segmentation of vast datasets into meaningful components such as trees, buildings, roads, and ground, the LiDAR point cloud processing technique empowers detailed spatial analysis and visualization, facilitating applications ranging from urban planning to ecological assessment. In urban construction, the segmentation of buildings and other structures aids in creating precise 3D models for planning and management. Similarly, in environment management, the semantic segmentation results of LiDAR point clouds assist in calculating vegetation cover and volume for biomass assessment.

Similar to image processing, Deep Learning (DL) is currently mainstream for LiDAR point cloud processing and analysis. However, due to the irregularity and unordered nature of LiDAR point clouds, they have more complicated structures than images. To achieve effective feature extracting and learning, several deep learning-based methods on point clouds have been proposed, which can be generally divided into four categories: Volume-based, Projection-based, Point-based methods, and Transformer-based methods.

Volume-based Methods. Inspired by image processing, VoxNet [80] proposed the 3D voxelization method for point cloud data processing. It used the regular volumetric grid to represent unstructured point clouds. After that, 3D convolutions can be directly performed for feature extraction. However, such methods initially struggled to leverage the sparsity of spatial point clouds due to rasterization. Additionally, constructing high-resolution voxelization models was computationally prohibitive because of the cubic growth in memory and processing requirements.

To address these issues, OctNet [97] introduced an unbalanced grid-octree algorithm, enabling the efficient representation of higher-resolution input data ($256 \times 256 \times 256$) compared to VoxNet. Subsequently, advancements such as 4D Spatio-temporal ConvNets [19] adopted sparse convolution algorithms, which significantly improved efficiency by avoiding computations in unoccupied voxel regions. These techniques effectively mitigated the computational and memory overheads associated with voxelization.

Despite these improvements, volume-based methods still face challenges. The fundamental problem of geometric information loss persists due to the transformation from irregular point clouds to regular 3D voxels, which can blur fine-grained details critical for certain tasks. While sparse convolutions and octree representations have alleviated memory and computational constraints, capturing local geometric relationships at very high resolutions remains challenging.

Projection-based Methods. As the pioneer of Projection-based methods, MVCNN [112] first projected point clouds into multiple views, transforming 3D data into 2D images. Then it used 2D Convolutional Neural Networks (CNNs) to extract the features from each view. After MVCNN, there have been several variants [129, 148] proposed to improve the

robustness and accuracy of the view feature combination. However, point cloud projection may result in the loss of geometric information. Additionally, it is still challenging to choose the optimal number and position of projection views for the underlying geometric structure modelling.

Point-based Methods. Point-based methods take the raw point cloud data with/without normals as input, dealing with unordered point clouds directly. PointNet [88] first used shared Multi-Layer Perceptrons (MLPs) to achieve point cloud feature learning. After that, PointNet++ [89] introduced local feature aggregation into PointNet, making the network aware of the local information. Inspired by the strong local feature extraction capability of CNNs, some further variants, such as PointCNN [57], PointConv [133], and DGCNN [127], proposed to design the 3D convolutional kernels and Graph Convolution algorithm to improve point cloud processing and analysis. Despite achieving great success in point cloud processing, the point-based methods still have some problems in feature learning: 1) it extracts only point-wise features or local features because of the locality inductive bias, making it challenging to learn global information. 2) The fixed weight matrices in MLPs are independent of the input data, which leads to the weak ability of feature representation. 3) MLPs are extremely sensitive to the input order because of the fixed weight matrices, not adaptive to the point cloud data.

Transformer-based Methods. In recent years, inspired by the application of Transformers in the image processing field, many Transformer-based point cloud processing methods have been proposed. The unique mechanism of Transformer, self-attention, overcomes the shortcomings of the MLP-based algorithm in point cloud processing and achieves excellent results. Point Cloud Transformer (PCT) [30] replaced the shared MLPs in PointNet architecture with Transformer blocks, making the network aware of long-range context dependencies. By doing this, PCT achieved better results than PointNet in terms of point cloud classification and segmentation. Point Transformer (PT) [160] applied the Transformer to local feature extraction, which has shown its superiority in feature aggregation. A detailed introduction to the Transformer structure is shown in in Section 1.1.2. And a detailed review of 3D Transformer methods is shown in Chapter 2.

1.1.2 3D Transformers

Transformers are now the dominant neural architecture in Natural Language Processing (NLP). Given the impressive ability to model long-range dependencies, they have been successfully adapted to the field of Computer Vision (CV) [32, 59, 136]. A standard Transformer encoder generally consists of six main components (Figure 1.1): 1) input (word)

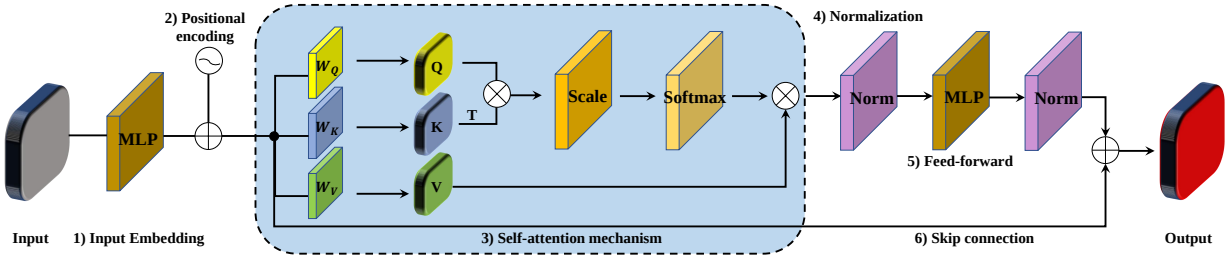


Figure 1.1: Illustration of the Transformer encoder architecture.

embedding; 2) positional encoding; 3) self-attention mechanism; 4) normalization; 5) feed-forward operation; and 6) skip connection. Compared to NLP and image processing, point cloud data presents more challenging characteristics, such as sparsity, uneven spatial distribution, and disorder. Firstly, for the issue of disorder, the self-attention mechanism enables the Transformer to be permutation-equivariant, which means the attention metrics generated by the self-attention mechanism depend on input data. Such flexibility allows the model to adapt to various input orders. Secondly, for the issue of sparsity and uneven distribution, traditional methods like CNNs, which rely on fixed local grids, struggle to handle these problems efficiently. However, the self-attention mechanism can establish the connection between any two input points, without depending on any extra gird structure. This allows Transformers to have strong long-range dependency modelling performance and dynamically attend to both dense and sparse regions, capturing important context from the entire point cloud. In addition, the positional encoding also enhances the sensitivity of the model to uneven distributions. Finally, for large-scale LiDAR point clouds, noises and outliers are also key drawbacks in point cloud processing. Transformers are able to dynamically adjust the attention weights in the self-attention mechanism to reduce the impact of noise and outlier points in global information modelling. By dividing the input features into different high-dimensional feature spaces, the multi-head self-attention mechanism aggregates features from multiple perspectives, which also improves the robustness of the model to noisy inputs.

To describe the Transformer architecture in more detail, let $\mathbb{P} = \{p_1, p_2, p_3, ..., p_N\} \in R^{N \times D}$ be an input point cloud, where D is the feature dimension of the input point. Typically in literature, "D equals to three" means only the 3D coordinate of each point is taken as input, while "D equals to six" means both the 3D coordinate and normal vector are taken as input. The details of the aforementioned encoder components are as follows.

Firstly, for the input embedding, P is projected to a high-dimension feature space which can facilitate subsequent learning. This can be achieved by MLP or other feature

extraction backbone networks like PointNet [88]. The embedded feature map is denoted as $\mathbb{X} \in R^{N \times C}$. Secondly, positional encoding is used to either capture the geometrical information, or the relative ordering of input tokens/points if relevant. Note that the Transformer is order-agnostic without this step, which is not an issue for point clouds, since they are naturally unordered. Nonetheless, frequency-based positional encoding can be used by mapping spatial coordinates with sine and cosine functions [120]. Moreover, there also exist learned position encoding schemes with a trainable parameter matrix B [160, 71], which are more adaptive to different input data. These positional encodings of spatial coordinates have been shown to benefit the learning of features at finer scales [115]. Thirdly, the core component of the Transformer encoder is the self-attention mechanism. Specifically, the embedded feature map $\mathbb{X}$ is projected to three different feature spaces using three learnable weight matrices $W_Q \in R^{C \times C_Q}, W_k \in R^{C \times C_K}, W_V \in R^{C \times C}$, where typically C_K equals C_Q. In this way, $Query$, Key, and $Value$ matrices are formulated as:

$$\begin{cases} Query & = \mathbb{X} \times W_Q, \\ Key & = \mathbb{X} \times W_K, \\ Value & = \mathbb{X} \times W_V, \end{cases} \tag{1.1}$$

where $\times$ represents matrix multiplication. Given the $Query$, Key, and $Value$ matrices, an attention map is formulated as:

$$Attentionmap = Softmax(\frac{Q \times K^T}{\sqrt{C_K}} + B), \tag{1.2}$$

where Q, K, V denote the $Query$, Key, and $Value$ matrices respectively. The attention map of the size of $N \times N$ measures the similarity of any two input points. It is also called the similarity matrix. Then the attention map and the matrix $Value$ are multiplied to generate the new feature map $\mathbb{F}$, of the same size as $\mathbb{X}$. Each feature vector in $\mathbb{F}$ is obtained by computing a weighted sum of all input features. It is therefore able to establish connections with all input features. When inputs are global, this process allows for the Transformer to easily learn global features. Therefore, compared with CNNs, Transformers excel at long-range dependency modelling because they rely on the self-attention mechanism, which enables direct interaction between all input points without being constrained by the inductive bias of locality inherent to CNNs. This makes Transformers particularly effective for capturing global context in point clouds, which is challenging for CNNs due to their reliance on fixed receptive fields. Moreover, Transformers are more adaptive to the sparsity of point clouds, as they do not require dense grid structures like CNNs, and can effectively process sparse data directly in its raw form. Compared with MLPs, Transformers offer two significant advantages. First, the attention map in Transformers is

dynamic and context-dependent during inference, allowing the model to adaptively focus on relevant features based on the input. This dynamic nature makes Transformers more flexible and robust compared to MLPs, which rely on fixed-weight matrices and cannot adapt to variations in input data. Second, the self-attention mechanism in Transformers is inherently permutation-equivariant, meaning that the model's output is consistent regardless of the order of the input points. In contrast, MLPs encode the order of inputs in their weight matrices, making them sensitive to input permutations and less suitable for unordered data like point clouds. Fourthly, a normalization layer is placed before and after the feed-forward layer, performing standardization and normalization on feature maps. There are two kinds of normalization methods used in this layer: LayerNormalization and BatchNormalization. The former is commonly used in NLP, while the latter is commonly used in CV like 2D or 3D data processing. Fifthly, a feed-forward layer is added to enhance the representation of attention features. Generally, it consists of two fully connected layers with a RELU function. Finally, a skip connection is used between the input and output of the self-attention module. There have been many self-attention variants using various skip connection forms [25, 140, 30] for performance improvement.

Considering the aforementioned advantages, Transformer-based point cloud processing and analysis is chosen as my research focus. However, despite the remarkable superiority of the Transformer in point cloud processing, introducing Transformers to the field of 3D point cloud processing still has several challenges. Based on the principle of the self-attention mechanism, the challenges could be divided into two main categories: **1) How to improve the model accuracy of the Transformer network in point cloud processing; 2) How to improve the model efficiency of the Transformer network, especially in large-scale LiDAR point cloud processing**. Therefore, the specific challenges are listed as follows:

(1) **Imbalance in local and global feature extraction.** The self-attention mechanism enables Transformers to emphasize global feature learning. However, it usually ignores critical local geometric details, which hinders the model's ability to undertake point cloud representation. Therefore, how to develop a Transformer framework that can consider both local and global feature learning is crucial to improving the point cloud representation ability.

(2) **High computational complexity and memory consumption.** The global self-attention mechanism in Transformers requires computing pairwise similarities between all points, leading to a computational complexity of $\mathcal{O}(N^2C)$. It results in significant computational and memory overhead, especially when processing large-scale LiDAR point clouds.

(3) **Limited exploration of efficient Transformer frameworks.** Most of the existing Transformer methods utilize the hierarchical structure, with Farthest Point Sampling (FPS) and k-Nearest neighbourhood (kNN) for point sampling and grouping. However, FPS is not only time-consuming but also only focuses on the geometric properties while ignoring the semantic features of point clouds. This causes deep learning models to downplay certain fine-level object parts with significant semantic information. Besides, it is usually challenging for the kNN searching strategy to generate semantically homogeneous neighbourhoods, thereby negatively impacting local feature aggregation. Recently, some superpoint-Transformer frameworks were proposed for efficient Transformer exploration, which utilizes the superpoint clustering strategy to reduce the input tokens to the model. However, traditional superpoint-based methods still have similar challenges with FPS and kNN searching. They rely on the initial geometric features for clustering, typically implemented as a pre-processing step. Such static approaches fail to dynamically adapt to semantic information during different network stages, leading to inefficiency and suboptimal performance in capturing fine-grained features.

1.2 Research Objectives

To address the aforementioned challenges in Transformer-based LiDAR point cloud processing, this thesis conducts the following three main studies:

(1) To improve the model accuracy of Transformers in LiDAR point cloud processing, it is necessary to address the issue of Transformers neglecting local information. Therefore, The first research objective of this thesis is to propose a general point cloud processing framework that can effectively capture and combine both local and global information, ensuring a comprehensive representation of LiDAR point clouds.

(2) To improve the model efficiency of the Transformer framework, it is essential to overcome the aforementioned shortages in existing hierarchical Transformer networks. Therefore, building upon the foundation laid by the first research work, the second objective of this thesis is to optimize the time-consuming point sampling and grouping processes, further designing a fast Transformer framework tailored to the representation of point clouds.

(3) To further reduce the computational and memory complexities of Transformers, it is important to address the high complexity $\mathcal{O}(N^2C)$ and substantial memory usage.

Considering that superpoint clustering significantly decreases the number of input tokens and thus reduces the computational overhead, incorporating the concept of superpoints into Transformer frameworks becomes a promising direction. However, traditional superpoint methods often exhibit limitations in adaptively representing semantic features. Therefore, the third research objective of this thesis is to design dynamic superpoint clustering and optimization strategies that further enhance the efficient Transformer framework, achieving state-of-the-art performance in both LiDAR point cloud processing accuracy and efficiency.

1.3 Thesis Structure

This thesis consists of six chapters, as shown in Figure 1.2.

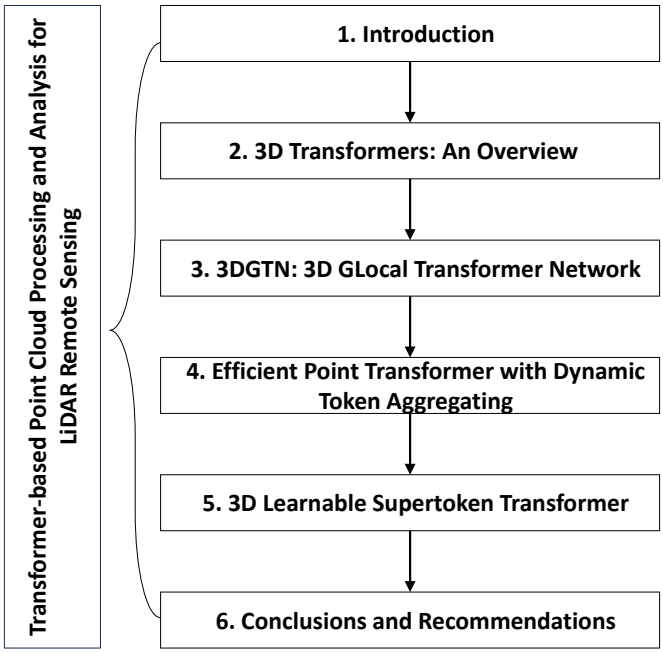


Figure 1.2: Structure of the thesis.

Chapter 1 introduces the background and motivation of the research, and then details the basic knowledge of the Transformer structure used in LiDAR point cloud processing. After that, three main challenges faced by introducing Transformers into point cloud processing are presented, and three research objectives are proposed correspondingly.

Chapter 2 reviews the related works of Transformers in point cloud processing, including Transformer-based point cloud classification and segmentation, efficient Transformer development, and self-attention mechanism variants.

Chapter 3 introduces a novel Transformer-based point cloud representational learning network, called **3D D**ual-attention **G**local **T**ransformer **N**etwork (**3DGTN**), to improve the point cloud representation performance of Transformers in both classification and segmentation tasks. The proposed framework is evaluated on classification and segmentation datasets, demonstrating its excellent performance on synthetic and LiDAR data.

Chapter 4 develops a fast point Transformer framework, named Trans**F**ormer with **D**ynamic **T**oken **A**ggregation (**DTA-F**ormer), for point cloud representation and processing, building upon the foundation laid by 3DGTN. Through the optimization and development of the framework, DTA-Former achieves much higher model latency while keeping excellent model accuracy.

Chapter 5 further optimizes the time and space complexity of DTA-Former, proposing a **3D L**earnable **S**upertoken **T**ransformer network (**3DLST**). By designing a novel dynamic supertoken optimization strategy, 3DLST achieves a further improvement in both model accuracy and efficiency. Extensive experiments demonstrate the SOTA performance of 3DLST compared to existing related works, as well as its strong adaptability to multi-source LiDAR point cloud data (airborne MS-LiDAR, aerial LiDAR).

Chapter 6 concludes the thesis and indicates the potential future research directions.

Chapter 2

3D Transformers: An Overview

Transformers have achieved great success in the field of point cloud processing. They have been applied to many common 3D tasks so far, such as classification, part segmentation, LiDAR point cloud segmentation, detection, registration, completion, etc. This chapter first makes a detailed introduction to 3D Transformers in point cloud classification and segmentation. Secondly, the development of efficient Transformer networks is reviewed, which is one of the current main development directions of Transformers. Finally, a series of self-attention mechanism variants are introduced.

2.1 Transformers in 3D Point Cloud Processing

As illustrated in Figure 2.1, Transformers have found extensive applications in 3D point cloud tasks such as classification, segmentation, detection, registration, and more. Among these, classification and segmentation account for the largest proportion. Accordingly, this thesis focuses primarily on these two tasks and provides an overview of the related work in this chapter.

2.1.1 3D Point Cloud Classification

There exist several Transformer-based point cloud classification methods, which can be divided into two categories: point-wise Transformer methods and channel-wise Transformer methods. Specifically, the former uses the self-attention mechanism to capture long- and/or short-range dependencies among input points. Therefore, they can be further divided

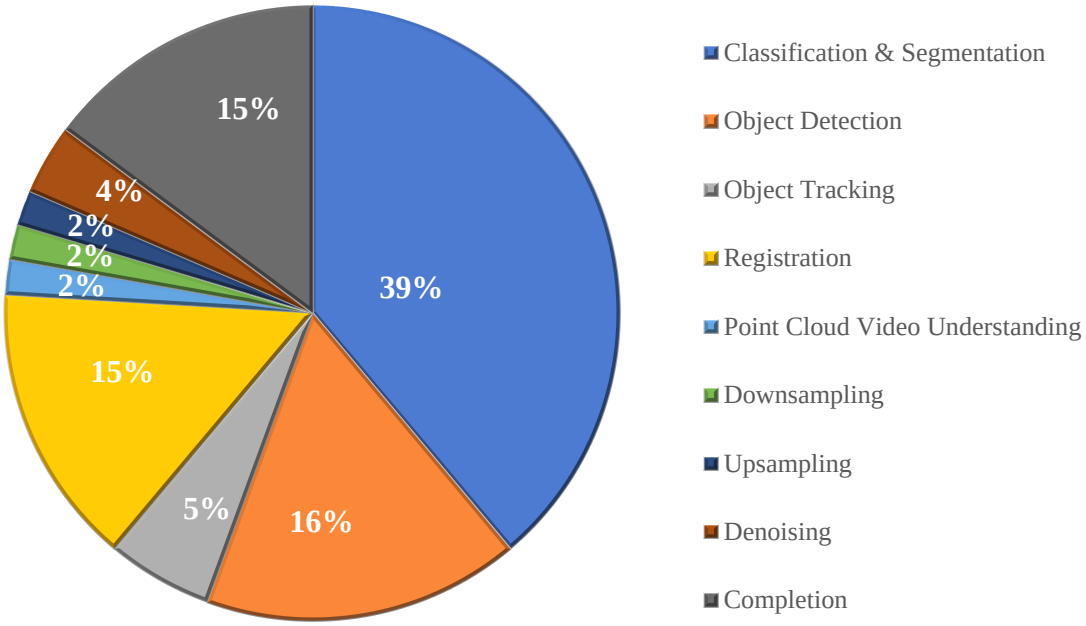


Figure 2.1: Applications of Transformers in 3D point clouds.

into global Transformer-based methods and local Transformer-based methods. However, a channel-wise Transformer, instead of operating in the point space, uses the self-attention mechanism to establish the relationship among different channels in the feature space.

PCT, as a standard global Transformer network, is introduced in [30]. In PCT, all input points are leveraged for global feature extraction. PCT first adopts a neighbourhood-embedding strategy to aggregate the local information, followed by feeding the embedded features into four stacked global Transformer blocks. Lastly, it utilizes a global Max and Average (MA) pooling to extract the global information for point cloud classification. 3CROSSNet proposed in [34] uses multi-scale global information for classification. Taking the raw point cloud as input, it first generates three point subsets with different resolutions by FPS. Secondly, it establishes kNN and extracts local information by a series of MLP modules for each point subset. Thirdly, the cascaded global Transformer blocks are applied to extract the global information of each subset. Lastly, given the multi-scale global features, 3CROSSNet uses the Cross-Level Cross-Attention (CLCA) and Cross-Scale Cross-Attention (CSCA) modules to capture long-range inter- and intra-level dependencies for classification.

As a local Transformer network, PT [160] focuses on extracting local information from

the Transformer. It has five local Transformer blocks operating on the downsampling point sets. Specifically, for each block, it constructs kNN neighbourhoods for sampling points, followed by utilizing a vector-attention mechanism to capture local features. After five local Transformer blocks, PT uses a global MA pooling to extract the global feature for classification. Local Feature Transformer Network (LFT-Net) [26] has a similar architecture. However, it uses an additional trans-pooling module to alleviate the feature loss during the pooling. Instead of searching neighbourhoods for sampling points in the spatial space, PointConT [64] explores the locality of points in the high-dimensional feature space, followed by computing the self-attention within each semantic-aware cluster.

The channel-wise Transformer proves to be effective in Geometric Back-projection Network (GBNet) [92]. GBNet uses a Channel-wise Affinity Attention (CAA) module to capture the channel-wise global information. It achieves state-of-the-art performance in point cloud classification. Dual Transformer Network (DTNet) [35] combines both the point-wise and channel-wise self-attention mechanisms for global feature extraction. The ablation experiments in DTNet show that the combined global feature works better than any feature from only point-wise or only channel-wise self-attention mechanism.

2.1.2 3D Point Cloud Segmentation

Many Transformer-based segmentation networks are developed from the corresponding classification networks. The segmentation network of PCT [30] takes its classification network variant as the encoder. The decoder first concatenates the pooled global feature with each point feature, enhancing the perception of global information for each point. Then the concatenated features are fed into a series of MLP layers for dense prediction, following PointNet [88]. Similarly, PT [160] also develops the segmentation network based on its classification framework. The authors design a U-net architecture for segmentation, where the decoder is symmetric to the encoder. Since it uses a hierarchical structure in the encoder, a transition-up module with trilinear interpolation is proposed in the decoder for point cloud upsampling. Further, Graph Transformer Network (GTNet) [165] integrates dynamic graph construction with Transformer architectures for point feature learning. It dynamically establishes local neighbourhoods based on deep features and employs both local and global Transformer modules to extract comprehensive point cloud features. Instead of using raw point clouds, Stratified Transformer [47] takes 3D voxels as input to the segmentation network. It applies Transformers in predefined local windows, following Swin Transformer [67]. To capture the global information and establish connections between different windows, it presents a novel *key* sampling strategy, enlarging the effective receptive field for each *query* point.

2.2 Efficient Point Transformers

Efficient point Transformers are a family of 3D Transformer models designed to improve the efficiency and scalability of traditional Transformer architectures while maintaining or even surpassing their performance on various tasks. Recently, many efficient point cloud Transformers [40, 151, 86, 113, 99, 121, 69] are proposed to reduce computation and memory costs and improve processing efficiency. The hierarchical processing framework is one of the most common techniques in point cloud processing, especially for large-scale LiDAR point clouds. PPT-Net [40] proposes a hierarchical encoder-decoder network to reduce the number of points gradually. Instead of using a pure Transformer architecture, it combines graph convolution-based [127] local feature embedding and Transformer-based global feature learning, which not only enhances long-term dependencies among points but also reduces the computational cost.

To reduce the number of input tokens and improve the models' ability to local feature extraction, PatchFromer [151], SPFormer [113], and SPT [99] all design superpoint-based local feature aggregation methods, based on positional and radiometric information from raw point clouds. They can generate geometrically-homogeneous point clusters for further local feature extraction, and only need to be calculated once, as a pre-processing step. Since the number of superpoints is much less than that of raw input points, superpoint-based Transformers achieve great progress in model efficiency improvement. SPFormer [113] designs a superpoint pooling layer based on pre-computed superpoints to guide point cloud downsampling, which significantly reduces the computational overhead of the following feature learning process. SPT [99] applies the static superpoint strategy to the input point clouds, generating a hierarchical superpoint structure by adjusting the granularity of superpoint clustering. It allows the model to exploit the context at different scales for semantic feature understanding, resulting in large receptive fields and high efficiency. However, the static point clustering strategy it uses cannot adaptively serve the semantic features extracted at different stages of the network, limiting its performance.

2.3 3D Self-attention Mechanism

Corresponding to the point-wise and channel-wise Transformers, the self-attention mechanisms can likewise be divided into point-wise and channel-wise self-attention. The attention

maps of these two kinds of Transformers can be expressed as:

$$\begin{aligned} \textit{Point}\text{-wise } \textit{Attn} &= Softmax\Big(\frac{Q \times K^T}{\sqrt{C_K}}\Big), \\ \textit{Channel}\text{-wise } \textit{Attn} &= Softmax\Big(\frac{Q^T \times K}{\sqrt{C_K}}\Big), \end{aligned} \tag{2.1}$$

where the size of *Point*-wise *Attn* is $N \times N$, while the size of *Channel*-wise *Attn* is $C_K \times C_K$. The softmax function is used to normalize the computed similarities into a probability distribution, thereby capturing the relative importance of each element. With the development of 3D Transformers, several self-attention variants are designed to improve algorithm performance.

2.3.1 Point-wise Self-attention

The point-wise self-attention is the most commonly used mechanism in 3D Transformers. Figure 2.2 shows the various existing point-wise self-attention mechanisms investigated in this thesis. Point Attention (P-A) network [25] (Figure 2.2(a)) and Attentional ShapeContextNet (A-SCN) network [140] (Figure 2.2(b)) design different residual structures in their Transformer encoders. The former strengthens the connection between the output and input of the module, while the latter establishes the relationship between the output and the $Value$ matrix of the module. Relevant experiments demonstrate that the residual connection is necessary in order to learn a good model [140].

Inspired by the Laplacian matrix $L = D - E$ in Graph Convolution Networks (GCNs) [10], where D is the diagonal degree matrix and E is the self-attention adjacency matrix, PCT [30] further proposes an Offset-Attention module (Figure 2.2(c)). It calculates the offset (difference) between the Self-Attention features and the input features X by matrix subtraction, which is analogous to a discrete Laplacian operation. Additionally, it refines the normalization of the similarity matrix by replacing *Scale* + *Softmax* with *Softmax* + L_1 *Norm* operation. In particular, the L_1 normalization re-scales each row or column so that the sum of the absolute values is 1, effectively sharpening the attention weights and reducing the influence of outliers. Based on the Offset-Attention, Zhou et al. [163] propose a Relation Attention Module (RAM) that has a similar structure as the OA module. The difference is that it first projects $Query$, Key, and $Value$ matrices into latent feature spaces by linear layers. Then, instead of generating the *Attentionmap* by multiplying the $Query$ and Key matrices directly, it applies L_2 normalization to the $Query$ and Key matrices. Compared with L_1, the L_2 normalization makes the sum of squared elements

equal to 1, thereby preventing the dominance of a few feature channels with extremely large magnitudes and distributing attention more evenly. Ablation experiments in [163] demonstrate that the L_2 normalization can improve the model performance.

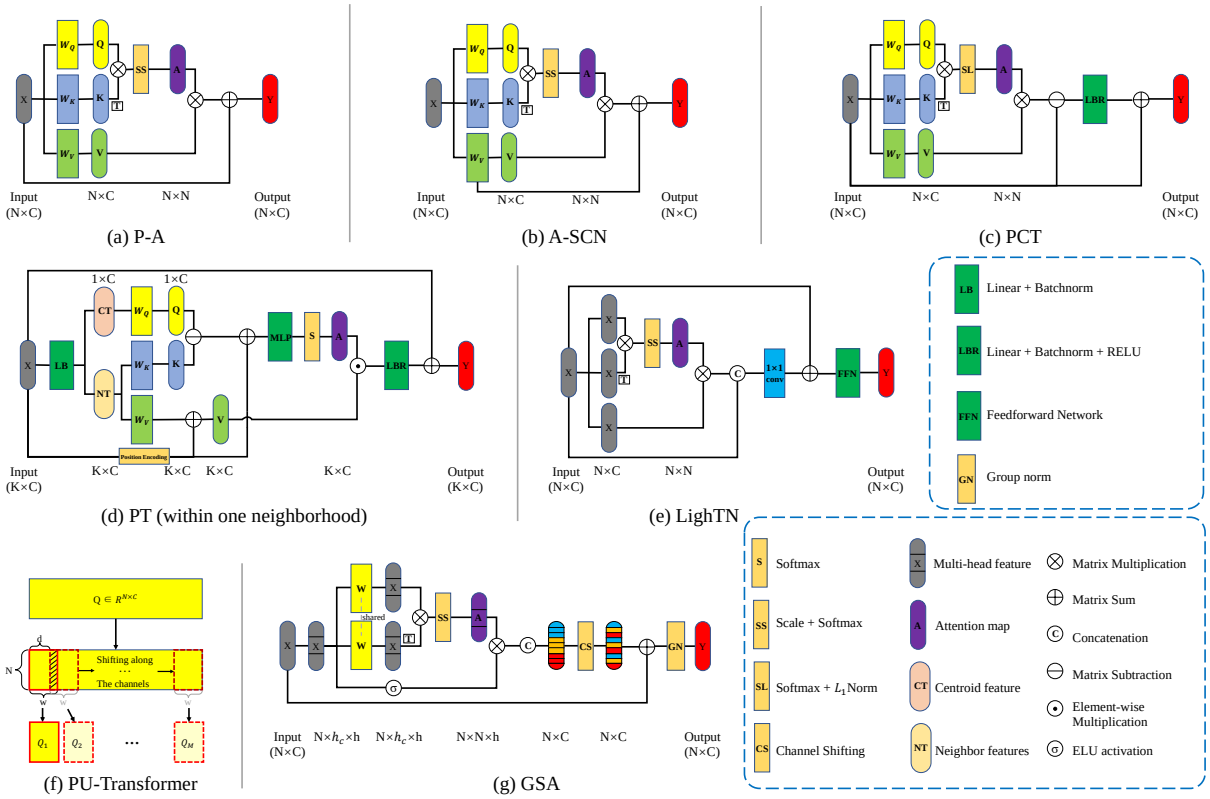


Figure 2.2: Architectures of Point-wise self-attention variants.

PT [160] (Figure 2.2(d)) introduces the vector subtraction attention operator to its Transformer network, replacing the commonly-used scalar dot-product attention. Compared with scalar attention, vector attention is more expressive since it supports adaptive modulation of individual feature channels, not just whole feature vectors. This kind of expression appears to be very beneficial in 3D data processing [160]. PT utilizes the subtraction-form vector attention to achieve local feature aggregation. The attention map is generated by merely building the connections between the centroid feature and its neighbour feature, instead of measuring the similarity between any two point features within a neighbourhood.

LighTN [124] presents a self-correlation module to reduce the computational cost. As

shown in Figure 2.2(e), it eliminates the projection matrices, W_Q, W_K, and W_V simultaneously in the self-attention mechanism. Only the input self-correlation parameters are used to generate attention features. According to Eq. 2.2, we see that the self-correlation mechanism generates a symmetry attention map $X \cdot X^T$, which satisfies the permutation invariance in point cloud processing [124]. The authors also conduct a series of ablation studies, removing different projection matrices, to demonstrate the effectiveness of the proposed self-correlation mechanism.

$$\begin{aligned} SA(X) &= FC_{out}(C(X)), \\ C(X) &= softmax\Big(\frac{X \times X^T}{\sqrt{C}}\Big) \times X, \end{aligned} \tag{2.2}$$

where $SA(*)$ represents the self-attention block, FC_{out} represents the linear transformation, $softmax(*)$ is the activation function, and C is the input feature dimension.

PU-Transformer [93] proposes a Shifted Channel Multi-head Self-Attention (SC-MSA) block to improve the MSA mechanism. Specifically, despite the rich information captured by MSA, only feature dependencies within the same head can be estimated, while lacking information propagation between different heads. To address this issue, as shown in Figure 2.2(f), PU-Transformer applies a window (dashed square) shift along the channels to ensure that any two consecutive splits have a uniform overlap area. Compared with the independent splits of the regular MSA, SC-MCA can enhance the channel-wise relations in the output features.

Group Shuffle Attention (GSA) proposed in [145] has two improvements compared with the standard MSA. The first one is that GSA is a parameter-efficient self-attention mechanism. It uses a shared projection matrix W to generate the $Query$ and Key matrices, and uses a non-linearity σ to generate the $Value$ matrix:

$$Attn_\sigma(X) = softmax\Big(\frac{Q \times K^T}{\sqrt{C}}\Big) \times \sigma(X), \tag{2.3}$$

where $Q = K = X \times W$ and C is the dimension of X. In this way, GSA can reduce the computational costs of the self-attention operation. The second one is that GSA introduces channel shuffle to MSA, which enhances the information flow between heads. As shown in Figure 2.2(g), unlike PU-Transformer [93], it re-groups the channels by rewriting each point feature.

2.3.2 Channel-wise Self-attention

The channel-wise self-attention mechanism draws more and more attention in the field of point cloud processing. Figure 2.3 shows the various existing channel-wise self-attention mechanisms investigated in this thesis. DT-Net [35] proposes a channel-wise MSA, applying the self-attention mechanism to the channel space. As shown in Figure 2.3(a), unlike the standard self-attention mechanism, channel-wise MSA multiplies the transposed *Query* matrix and *Key* matrix. By this means, the attention map is generated to measure the similarities between different channels, as described in Eq. 2.1.

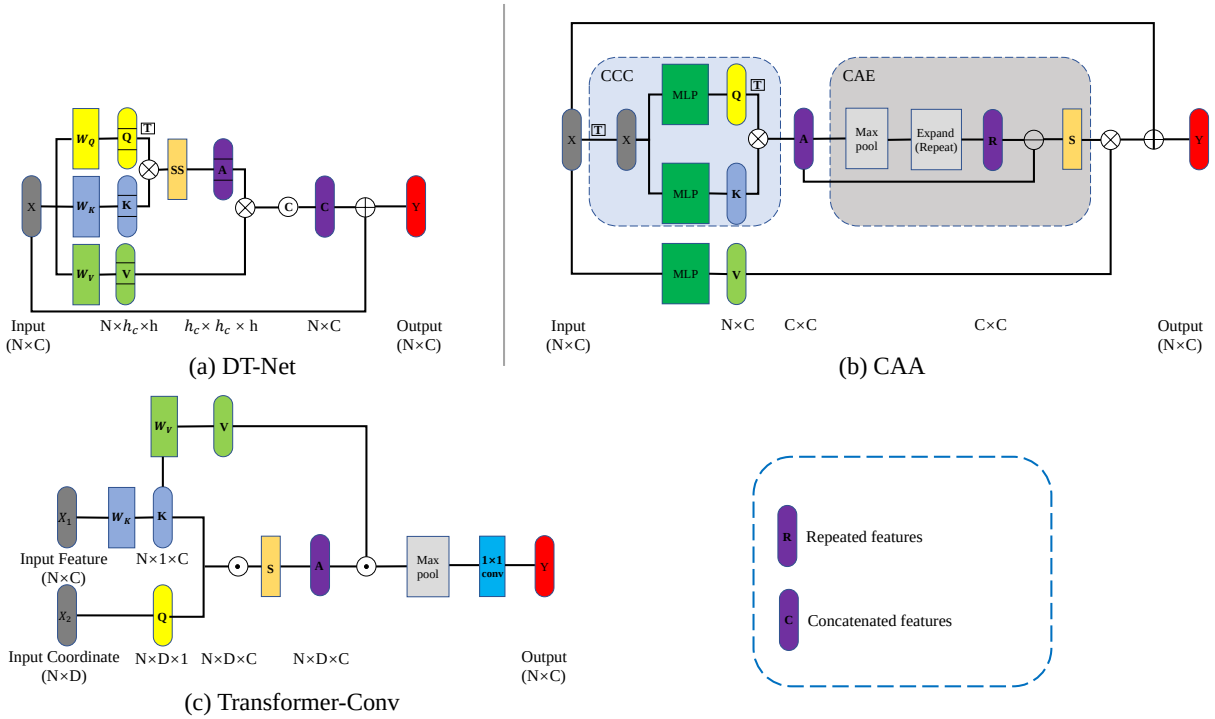


Figure 2.3: Architectures of Channel-wise self-attention variants.

As shown in Figure 2.3(b), the CAA module [92] utilizes a similar approach—a Compact Channel-wise Comparator block (CCC)—to generate the similarity matrix between different channels. Moreover, it designs a Channel Affinity Estimator block (CAE) to generate the affinity matrix, strengthening the connection between distinct channels and avoiding aggregating similar/redundant information. The *Value* matrix is generated by an MLP layer, and the final feature map is obtained by multiplying the affinity matrix and

$Value$ matrix. Additionally, the CAA module uses a regular skip connection between the input and output feature map.

The Transformer-Conv module proposed in [141] is used to learn the potential relationship between feature channels and coordinate channels. As shown in Figure 2.3(c), the $Query$ matrix and Key matrix are generated by coordinates and features of the point cloud, respectively. Then the similarity matrix is produced by a relation function β (e.g., element-wise multiplication) and channel softmax operation. Different from the above methods, the $Value$ matrix in the Transformer-Conv module is generated from the Key matrix by linear projection. This operation can establish a response relationship between the $Value$ matrix and similarity matrix. Such a relationship is captured by multiplying the similarity matrix and $Value$ matrix in an element-wise manner. Lastly, the final feature map is generated by using channel max-pooling and further 1×1 convolution for controlling the output dimensionality in an efficient manner.

2.4 Chapter Summary

This chapter presents a review of 3D Transformers in classification and segmentation, efficient Transformer development, and self-attention mechanism variants. By reviewing the previous 3D Transformer works, some challenges faced by 3D Transformer researchers are pointed out in this section. Firstly, previous 3D Transformer methods tend to only consider global or local information extraction. This issue makes it still challenging for 3D Transformer methods to capture the global information of the target accurately while preserving the local features. Secondly, despite the great success of Transformers in point cloud processing, its massive linear operations tend to result in high computational complexity and memory consumption. Finally, most hierarchical Transformers performed traditional point cloud downsampling and neighbourhood searching algorithms, which are very time-consuming and only operated in the spatial space. Such methods tend to hinder semantic information extraction and aggregation in the deep feature space. Besides, although the recent superpoint-based Transformer methods achieved great progress in model efficiency improvement, the static superpoint clustering strategy fails to adaptively serve the deep semantic features, limiting their performance. All of these are main challenges faced by 3D Transformer research, which are also the research focuses of this thesis.

Chapter 3

3DGTN: 3D GLocal Transformer Network[1]

3.1 Introduction

Although Transformers have demonstrated strong feature learning capabilities in LiDAR point cloud applications, they still have limitations in terms of effectively modelling both local information and global information. This chapter introduces a 3D 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. 3DGTN is tailor-designed to improve combined global and local feature learning in 3D point cloud data processing, with the following key characteristics.

(1) A Global Feature Learning (GFL) block with the dual-attention mechanism is designed to efficiently learn the global context information from multiple perspectives. The dual-attention mechanism integrates Point-wise Self-Attention (PSA) and Channel-wise Self-Attention (CSA) to improve the learning of critical information in both the spatial domain and feature domain.

[1]The content of the chapter has been published in IEEE Transactions on Geoscience and Remote Sensing with the paper entitled "3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation". The code has been released on https://github.com/d62lu/3DGTN.

(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 GCN for accurate 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.

3.2 Methodology

This section introduces the encoder and decoder structures of 3DGTN for both point cloud classification and segmentation. The pipeline of the method is shown first, and then the main blocks in the encoder and decoder are introduced respectively.

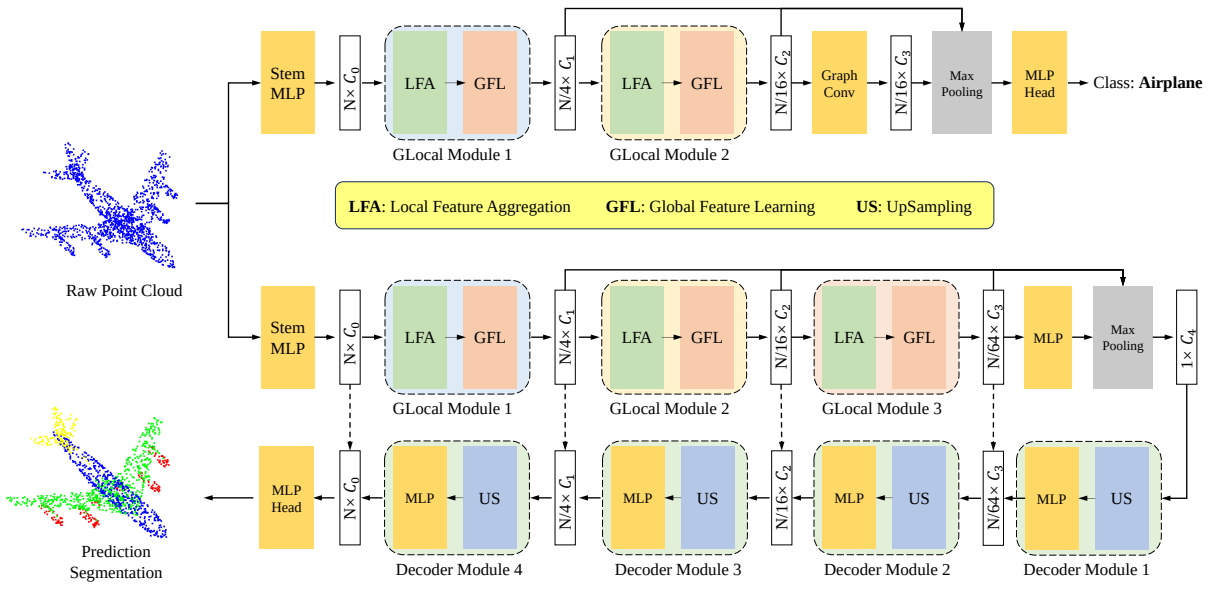


Figure 3.1: 3DGTN networks for point cloud classification and segmentation.

3.2.1 Overview

Figure 3.1 shows the overall pipeline of 3DGTN. The 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.A stem MLP block is designed first to project the input data into a higher-dimensional space. After that, the projected features are fed into several GLocal modules consisting of stacked LFA and GFL blocks, processed in a hierarchical manner for GLocal feature extraction. Specifically, the LFA block is adapted from the multi-scale GCN [127], and the GFL block is adapted from the Transformer. Following this, the max-pooling operation is used on the output feature maps of each module, to obtain the GLocal feature of each level. Then, we concatenate them for multi-level GLocal feature generation. Given the extracted features, an Multi-Layer Perceptron (MLP) head is leveraged 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, an ALL-MLP decoder structure is adopted, 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 used in 3DGTN has been widely applied to hierarchical networks of point cloud processing. It generates new points by considering the weighted averages of neighbouring points in the geometric space as 3D linear interpolation, providing an effective method to enhance the density and precision of 3D data representations.

3.2.2 Local Feature Aggregation (LFA) Block

The LFA block is designed for local feature aggregation, which is introduced as follows. As shown in Figure 3.2, the input point cloud is first downsampled to $N/4$ points via FPS [89], generating a sampled point subset $\mathbb{S}$, where N is the number of the input points. After that, the LFA block constructs multi-scale neighbourhoods by query ball searching [89] for each sampled point, to ensure the diversity of the receptive fields. In each neighbourhood χ_i of the sampled point S_i, a fused feature $\mathbb{C}_{ij}$ is generated by computing the difference between the j-th neighbourhood point χ_{ij} in χ_i and S_i:

$$\mathbb{C}_{ij} = concat(\mathbb{F}_{ij} - \mathbb{F}_i, \mathbb{F}_i), \tag{3.1}$$

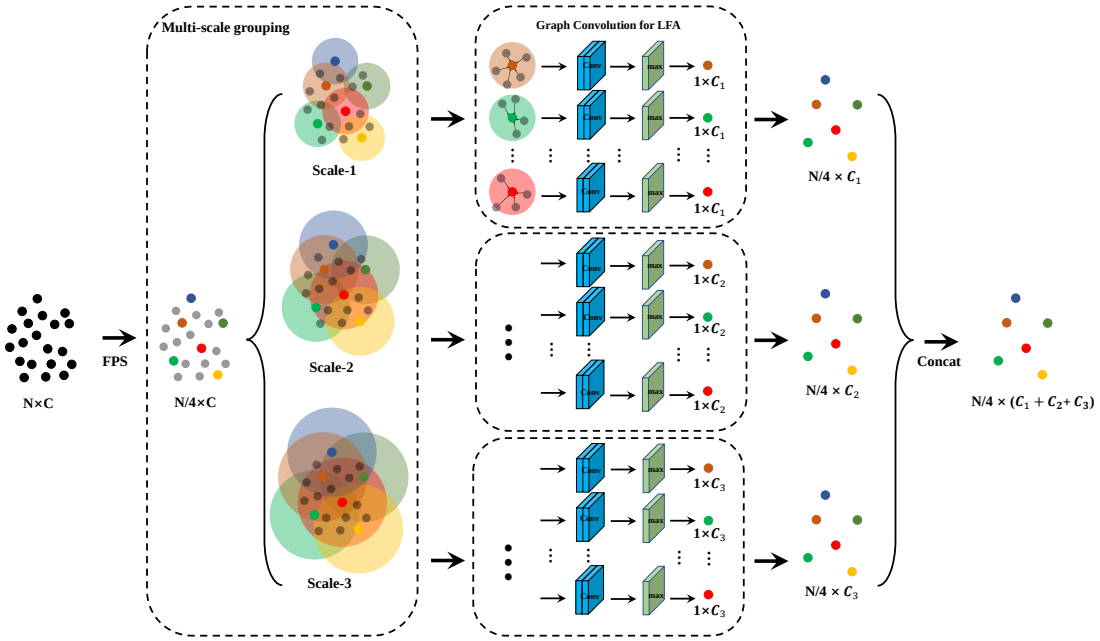


Figure 3.2: Multi-scale Local Feature Aggregation (LFA) block.

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

$$l_i = \underset{j \in \chi_i}{maxpooling}\,(Conv(\mathbb{C}_{ij})), \tag{3.2}$$

where $l_i \in R^{1\times C}$ is the aggregated local feature of S_i, $Conv$ is a convolution operation with 1×1 kernels. $Conv(\mathbb{C}_{ij})$ establishes semantic relationships between the sampling point S_i and neighbourhood point χ_{ij}. As such, a neighbourhood feature set of S_i (denoted as $\mathbb{N}_i$) containing local information is generated. Then, the max-pooling operation is used to aggregate the local information to S_i, denoted as l_i.

The multi-scale local feature L_i of S_i, can be expressed as via a concatenation as:

$$L_i = concat(l_{i1}, l_{i2}, l_{i3}), \tag{3.3}$$

where $l_{i1} \in R^{1\times C_1}, l_{i2} \in R^{1\times C_2}, l_{i3} \in R^{1\times C_3}$ represent three local features of S_i at three different scales.

3.2.3 Global Feature Learning (GFL) Block

The GFL block contains two kinds of self-attention mechanisms: PSA and CSA. PSA is proposed to build the spatial relationship between points, achieving long-range context dependency modelling. CSA is utilized to measure the correlation among different feature channels. It can improve context information modelling by highlighting the role of interaction across various channels. By combining these two kinds of self-attention mechanisms, the GFL block can capture global features from multiple perspectives. A detailed introduction to these two mechanisms is as follows.

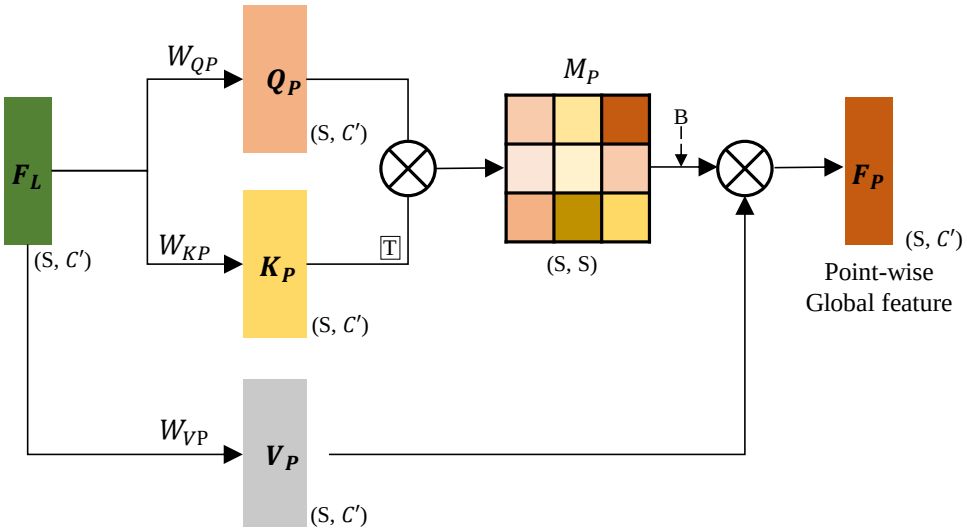


Figure 3.3: Point-wise Self-Attention mechanism.

$\mathbb{F}_{\mathbb{L}} = \{L_i\}_{i \in S} \in R^{S \times C'}$ from the LFA block is taken as input, where S is the number of sampled points in $\mathbb{S}$, and $C' = C_1 + C_2 + C_3$ denotes the feature dimension of $\mathbb{F}_{\mathbb{L}}$. As shown in Figures 3.3 and 3.4, PSA and CSA have similar algorithm flows. Specifically, taking the sampling point set $\mathbb{S}$ as input, I first project it into three different feature spaces to generate $Query$, Key, and $Value$ matrices:

$$\begin{aligned} Q_P &= \mathbb{F}_{\mathbb{L}} \times W_{QP}, \\ K_P &= \mathbb{F}_{\mathbb{L}} \times W_{KP}, \\ V_P &= \mathbb{F}_{\mathbb{L}} \times W_{VP}, \end{aligned} \tag{3.4}$$

where Q_P, K_P, and V_P represent $Query$, Key, and $Value$ matrices in PSA, W_{QP}, W_{KP}, and W_{VP} are learnable weight matrices. For the PSA, the attention map $M_P \in R^{S \times S}$ can

be formulated as:

$$M_P = softmax(\frac{Q_P \times K_P^T}{\sqrt{C'}} + B), \tag{3.5}$$

where B is a learnable position encoding matrix defined by [160]. M_P and V_P matrices are multiplied to generate the feature map $\mathbb{F}_\mathbb{P}$ as the output of PSA, of the same size as $\mathbb{F}_\mathbb{L}$.

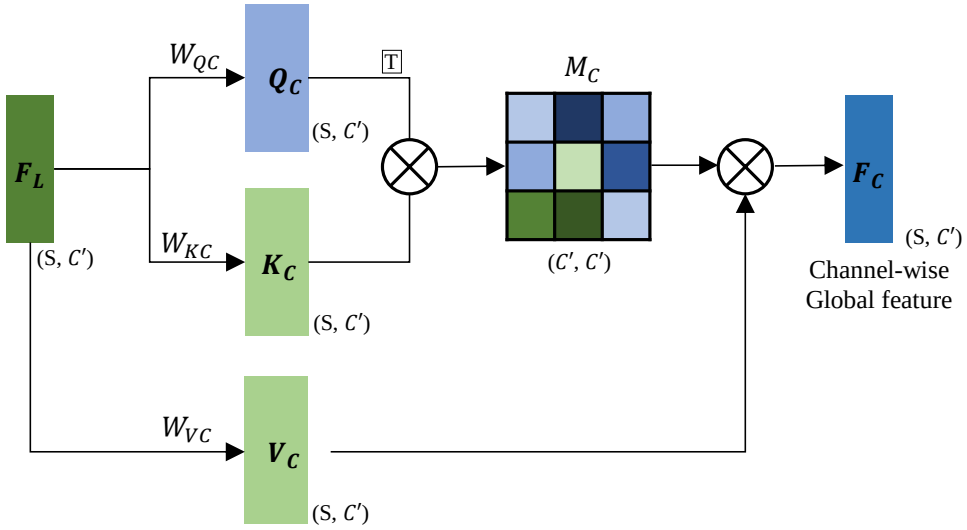


Figure 3.4: Channel-wise Self-Attention mechanism.

Similarly, for the CSA, the $Query$, Key, and $Value$ matrices can be expressed as:

$$\begin{aligned} Q_C &= \mathbb{F}_\mathbb{L} \times W_{QC}, \\ K_C &= \mathbb{F}_\mathbb{L} \times W_{KC}, \\ V_C &= \mathbb{F}_\mathbb{L} \times W_{VC}, \end{aligned} \tag{3.6}$$

where Q_C, K_C, and V_C represent $Query$, Key, and $Value$ matrices in CSA, W_{QC}, W_{KC}, and W_{VC} are learnable weight matrices. Then the attention map $M_C \in R^{C' \times C'}$ can be formulated as:

$$M_C = softmax(\frac{K_C^T \times Q_C}{\sqrt{C'}}). \tag{3.7}$$

V_C and M_C matrices are multiplied to generate the new feature map $\mathbb{F}_\mathbb{C}$ as the output of CSA, of the same size as $\mathbb{F}_\mathbb{L}$. Given global feature maps $\mathbb{F}_\mathbb{P}$ and $\mathbb{F}_\mathbb{C}$, we combine them by the element-wise addition:

$$\mathbb{F}_\mathbb{G} = \mathbb{F}_\mathbb{P} + \mathbb{F}_\mathbb{C}. \tag{3.8}$$

Lastly, a skip connection between $\mathbb{F}_{\mathbb{G}}$ and the input feature set $\mathbb{S}$ is applied:

$$\mathbb{F}_{\mathbb{G}} = \mathbb{F}_{\mathbb{L}} + LBR(\mathbb{F}_{\mathbb{G}}), \tag{3.9}$$

where $\mathbb{F}_{\mathbb{G}}$ is the final global feature map, and LBR denotes the combination of $Linear$, $BatchNorm$, and $ReLU$.

3.3 Experimental Results and Discussion

This section first introduces the implementation details of 3DGTN, including hardware configuration and hyperparameter settings. Secondly, the performance evaluation of the method on classification and segmentation tasks is presented, compared to existing deep learning-based methods, including Transformer-based methods. Specifically, 3DGTN is first tested on the widely-used ModelNet40 and ScanObjectNN datasets [135, 118] for the classification task. Secondly, for object part segmentation, 3DGTN is tested on the ShapeNet dataset [146]. Thirdly, semantic segmentation is tested on the challenging large-scale MS-LiDAR dataset [161]. Finally, the ablation experiment results on the main components of the method are presented.

3.3.1 Implementation Details

The classification and segmentation networks were implemented in PyTorch. Both were trained and tested on an NVIDIA Tesla V100 GPU. I 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. I trained classification, part segmentation, and semantic segmentation networks for 250, 300, and 500 epochs respectively, with the same batch size of 16. To make a fair comparison, the experiments on model efficiency in this thesis are all tested on an NVIDIA GeForce RTX 2080 GPU.

3.3.2 Point Cloud Classification

Datasets and Metrics. The ModelNet40 dataset [135] contains 12,311 CAD models across 40 object categories, introduced for 3D shape classification. Each category includes commonly encountered objects (e.g., chairs, tables, desks), covering a wide range of geometric variations. All CAD models are consistently oriented and have been widely used

to benchmark the performance and generalization capability of 3D deep learning methods. Following [89], the dataset is split into 9,843 training samples and 2,468 testing models. For a fair comparison, each input point cloud is downsampled to 1,024 points with normals via FPS. Since these point clouds are generated from 3D meshes, each point's normal can be obtained from the corresponding surface normal. The overall accuracy (OA) is adopted to evaluate classification performance, formulated as:

$$OA = \frac{T}{N}, \tag{3.10}$$

where T is the number of 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 total number of classes, N is the total number of point clouds, and $N = \sum_{i=1}^{K} N_i$. Additionally, we measure the latency (ms) to assess model efficiency, referring to the time required for a single forward pass to generate predictions.

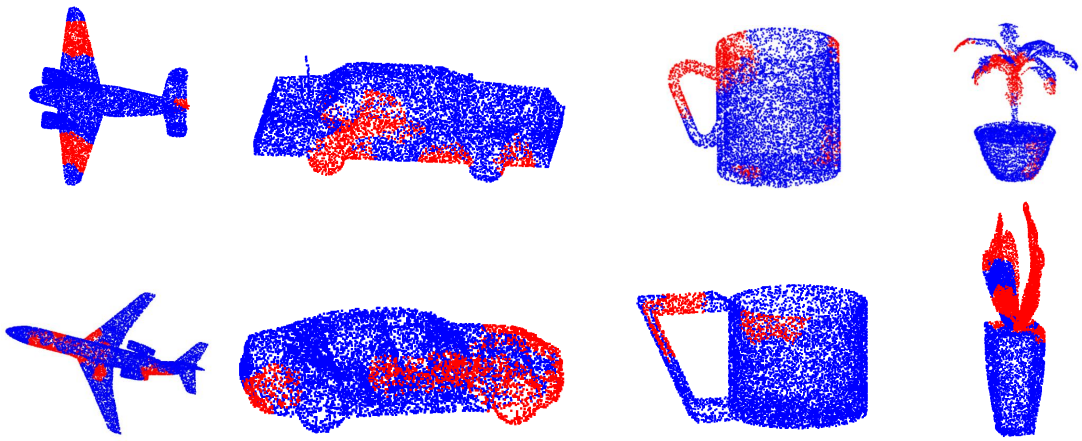
Table 3.1: Quantitative comparison of 3DGTN on the ModelNet40 dataset.

Methods	Input Size	OA (%)	Latency (ms)
Other Learning-based Methods			
PointNet++[89]	1024	91.9	35.3
PointCNN[57]	1024	92.2	221.2
DGCNN[127]	1024	92.2	85.8
DeepGCN[51]	1024	93.6	81.2
PointNeXt[90]	1024	93.7	32.7
RepSurf[95]	1024	**94.0**	99.5
Point-PN[155]	1024	93.8	45.2
APES[131]	1024	93.8	**14.5**
Transformer-based Methods			
PointASNL[144]	1024	93.2	923.6
PCT[30]	1024	93.2	92.4
PT[160]	1024	93.7	530.2
GBNet[92]	1024	93.8	103.7
PatchFormer[151]	1024	93.5	**34.3**
PointConT[64]	1024	93.5	79.1
3DGTN [70]	1024	**94.0**	66.7

Performance Comparison. As shown in Table. 3.1, 3DGTN is compared with the existing Transformer-based methods and other deep learning-based methods. Specifically,

3DGTN achieves the best result of OA (94.0%) among all benchmarked methods, outperforming the prior SOTA (PointNext [90]) during that period. For the model efficiency, 3DGTN is second only to the SOTA efficient Transformer during that period, PatchFormer [151]. Due to the naive implementation of several time-consuming operations in 3DGTN such as downsampling and kNN neighbourhood construction, the inference speed of the method can still be improved. Please refer to Appendix A for more comparison results of 3DGTN in the classification task.

Visualization. The Grad-CAM [103] map visualization results are generated 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 Figure 3.5, we obtain the regions of interest of the network for several point clouds of the Airplane, Car, Cup, and Plant classes. From the results, the attention (coloured 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 can be seen, all the regions of interest are consistent with the human visual system, which helps us establish appropriate trust in predictions from deep networks.



(a) Airplane (b) Car (c) Cup (d) Plant

Figure 3.5: Visualization of 3DGTN attention on the ModelNet40 classification dataset.
© 2022, IEEE

3.3.3 Part Segmentation on ShapeNet Dataset

Dataset and Metrics. ShapeNet Dataset and Metrics. The ShapeNet dataset contains 16,880 3D models spanning 16 shape categories, split into 14,006 training samples and 2,874 testing models, following PT [160]. Each 3D shape is annotated with at least two parts, leading to 50 part labels in total. These shapes are also widely used for benchmarking part segmentation tasks, covering a variety of objects such as chairs, airplanes, and lamps. To ensure consistency, each input point cloud is downsampled to 2,048 points with normals via FPS. The metrics of mIoU (%) and latency (ms) are used for performance evaluation, where the detailed definition of mIoU is shown in Appendix A.2.

Table 3.2: Part segmentation results of 3DGTN on the ShapeNet dataset.

Methods	Input Size	mIoU (%)	Latency (ms)
Other Learning-based Methods			
PointNet++(MSG)[89]	2048	85.1	47.7
PointCNN[57]	2048	86.1	252.1
DGCNN[127]	2048	85.2	96.2
DeepGCN[51]	2048	86.3	105.6
RepSurf[95]	2048	86.4	117.1
Point-PN[155]	2048	**86.6**	58.9
APES[131]	2048	85.8	**24.8**
Transformer-based Methods			
PointASNL[144]	2048	85.4	1023.2
PCT[30]	2048	86.4	101.1
PT[160]	2048	86.6	560.2
GBNet[92]	2048	85.9	198.4
Stratified Transformer[47]	2048	86.6	1792.6
PatchFormer[151]	2048	86.5	**45.8**
Li et al.[58]	2048	86.0	-
Hassan et al.[38]	2048	86.3	-
3DGTN[70]	2048	**86.6**	136.8

Performance Comparison. The comparison results are shown in Table. 3.2. As measured by mIoU, 3DGTN achieves competitive results (86.6%) compared with the SOTA Transformer-based methods during that period such as Stratified Transformer [47]. This demonstrates the excellent performance of 3DGTN in terms of part segmentation. Several part segmentation results are shown in Figure 3.6.

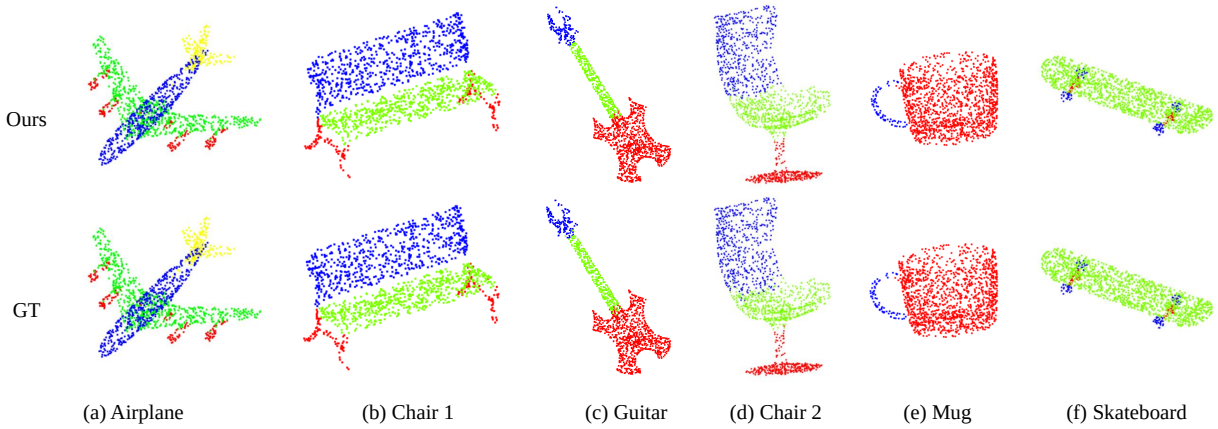


Figure 3.6: Part segmentation results of 3DGTN from the ShapeNet dataset.
© 2022, IEEE

3.3.4 Semantic Segmentation on Airborne MS-LiDAR Dataset

Dataset and Metrics. The airborne MS-LiDAR dataset, introduced by [161], originates from the Teledyne Optech Titan LiDAR system. The data spans 13 areas located in a Canadian town, with Areas 1–10 designated for training and Areas 11–13 for testing. Specifically, each area spans over 15,0000 m^2 and has an average point density of approximately 3.6 points/m^2. Each point includes six attributes: XYZ coordinates, MIR (1550 nm wavelength), NIR (1064 nm wavelength), and Green (532 nm wavelength). The dataset includes over 10 million labeled points, distributed across six semantic categories, as shown in Table 3.3. It poses a challenge due to its imbalanced class distribution. The quantity of road points exceeds the number of powerline points by more than 30 times. The MS-LiDAR dataset serves as a valuable resource for benchmarking deep learning algorithms in applications such as land cover classification, urban planning, and environmental monitoring. We adopted the data pre-processing method in [161] for training/testing sample generation in our experiments, to ensure a fair comparison. Each area was divided into a group of local blocks using k-Nearest neighbour (kNN) searching, with k set to 4096. The average F_1 score [137], mIoU, and OA are used for performance evaluation. Please refer to Appendix A.2 for more details of the metric definition.

Performance Comparison. As shown in Table. 3.4, the semantic segmentation results of airborne MS-LiDAR data are presented in the form of a confusion matrix. Since the CSA mechanism is integrated into global feature learning, 3DGTN can handle MS LiDAR point cloud segmentation well. Specifically, from the table, in the case of extremely

Table 3.3: Comparison of the point numbers in each category in the MS-LiDAR dataset.

Class	Trainset (No. 1-10)	Testset (No. 11-13)
Road	586,987	258,942
Building	415,763	116,085
Grass	2,577,194	983,724
Tree	2,623,317	863,519
Soil	124,566	33,214
Powerline	12,155	7,545

imbalanced data, 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 road, which tends to confuse the network, the segmentation results of the soil are not very satisfactory. The comparison results are shown in Table. 3.5[2]. As can be seen, 3DGTN outperforms all benchmarked methods in terms of average F_1 score (88.6%). It surpasses the prior SOTA methods such as [137] and [154] by 5.3 and 0.5 absolute percentage points, respectively. It also achieves the best OA (95.2%) and a competitive mIoU (82.1%). The prediction results of testing data are shown in Figure 3.7. These results demonstrate that 3DGTN has excellent performance in processing real-scanned data.

Table 3.4: Confusion matrix (%) of 3DGTN on the airborne MS-LiDAR dataset.

Categories		**True Label**					
		Road	Building	Grass	Tree	Soil	Powerline
Prediction Label	Road	90.5	3.2	0.0	0.1	15.1	0.0
	Building	6.6	93.9	0.8	0.6	23.9	0.2
	Grass	0.0	1.2	98.9	2.3	0.2	4.4
	Tree	0.0	0.1	0.1	97.1	0.4	0.0
	Soil	2.8	1.6	0.0	0.0	60.4	0.0
	Powerline	0.0	0.0	0.1	0.0	0.0	95.4
Precision		84.5	96.3	98.5	98.1	56.4	94.1
Recall		90.5	93.9	98.9	97.1	60.4	95.4
F_1		87.4	95.1	98.7	97.6	58.3	94.7

I also explored the importance of different channels in the MS-LiDAR data. Specifically,

[2]Methods marked with * were not originally tested on this dataset in their respective papers. The reported results were obtained by running the official code provided by the authors.

Table 3.5: Quantitative comparison of 3DGTN on the airborne MS-LiDAR dataset.

Methods	Average F_1 score (%)	mIoU (%)	OA (%)	Latency (ms)
PointNet++ [89]	72.1	58.6	90.1	322.6
DGCNN [127]	71.6	51.0	91.4	86.2
RSCNN [66]	73.9	56.1	91.0	158.7
GACNet [122]	67.7	51.0	90.0	277.8
AGConv [164]	76.9	71.2	93.3	312.5
SE-PointNet++ [43]	75.9	60.2	91.2	-
FR-GCNet [161]	78.6	65.8	93.6	-
PT* [160]	80.5	73.6	93.1	285.7
PPT-Net* [40]	80.1	73.6	92.7	43.3
PatchFormer* [151]	82.4	77.8	93.1	62.9
Xiao et al. [137]	83.3	79.3	94.0	-
ResMRGCN-28* [52]	81.1	74.0	93.3	45.7
GCNAS [154]	88.1	**82.3**	95.2	-
3DGTN	**88.6**	82.1	**95.2**	217.4

I removed each of the three channels (MIR, NIR, $Green$) of MS-LiDAR data, and then analyzed the corresponding performance changes in Table. 3.6. When the NIR channel was removed (Row 4), the performance dropped significantly (average F_1 score was reduced to 74.3% from 88.6%). 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.

Table 3.6: Results (%) comparison of input data with the removal of different channels on the airborne MS-LiDAR dataset.

Input				Road	Building	Grass	Tree	Soil	Powerline	Average F_1	mIoU	OA
XYZ	MIR	NIR	Green									
✓	-	✓	✓	84.1	90.7	95.8	96.2	51.2	91.1	84.9	79.9	93.9
✓	✓	-	✓	60.6	84.6	93.3	90.6	33.3	83.4	74.3	69.8	86.4
✓	✓	✓	-	83.1	90.8	92.7	97.0	45.2	88.1	82.8	78.7	92.1
✓	✓	✓	✓	**87.4**	**95.1**	**98.7**	**97.6**	**58.3**	**94.7**	**88.6**	**82.1**	**95.2**

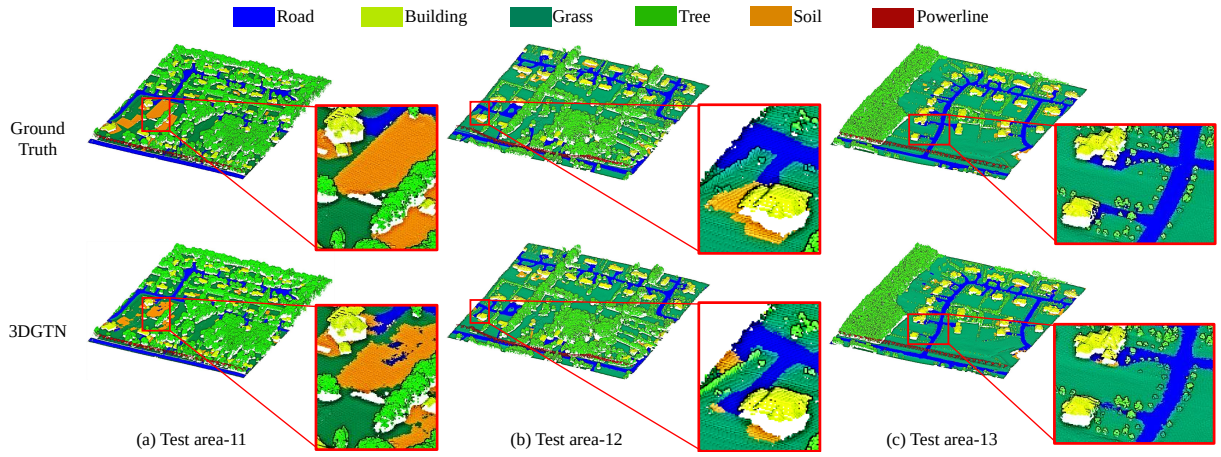


Figure 3.7: Semantic segmentation results of 3DGTN on the airborne MS-LiDAR dataset.

3.3.5 Ablation Study

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

Local Feature Aggregation Block. The effectiveness of the LFA block was first investigated, which is used to capture local information. As shown in Table. 3.7 Row 2, the performance with the MLP-based LFA block is 93.1% in terms of OA, which is lower than that with the GCN-based LFA block (94.0%). This demonstrates that the GCN-based LFA block plays an important role in the proposed algorithm. I also replaced the multi-scale strategy of the LFA block with the single-scale one. As shown in Table. 3.7 Row 3, the classification performance of the single-scale strategy (92.9%) is lower than the multi-scale one (94.0%). This suggests that the multi-scale features are beneficial to enhancing the expression of local information, thereby improving the performance of the algorithm. Finally, I replaced the FPS method with random sampling, to investigate the performance of 3DGTN with different sampling approaches. As shown in Table. 3.7 Row 4, the classification accuracy dropped slightly with random sampling (93.5% in terms of OA). This is because compared with random sampling, FPS could maintain the geometric characteristics of the target point cloud better. However, as measured by latency, 3DGTN with FPS (66.7ms) is more time-consuming than that with random sampling (43.5ms).

Global Feature Learning Block. I conducted a detailed ablation study on the GFL block. As shown in Table. 3.7, when the GFL block was removed, the performance

Table 3.7: Ablation studies for 3DGTN on the ModelNet40 dataset.

Ablation		OA (%)	Latency (ms)
Local feature aggregation	Graph convolution → Standard MLP	93.1	58.8
	Multi-scale → Single-scale	92.9	52.6
	FPS → Random sampling	93.5	43.5
Global feature learning	–	92.7	58.8
	– CSA	93.7	62.5
	– PSA	93.2	62.5
3DGTN		**94.0**	66.7

dropped significantly, which demonstrates that the GFL block is essential to the algorithm. Secondly, since the GFL block contains two important mechanisms: PSA and CSA, I also studied the effectiveness of each mechanism. When the CSA was removed, the classification accuracy (OA) dropped from 94.0% to 93.7%. Likewise, when the PSA was removed, there was a similar drop (from 94.0% to 93.2%). These results suggest that both self-attention mechanisms are effective in improving classification performance.

3.4 Chapter Summary

In this chapter, we propose a general Transformer-based point cloud representation network, dubbed 3DGTN, to enhance feature learning for both classification and segmentation tasks. The method comprises three main components. First, we design a GFL block that incorporates a dual-attention mechanism (PSA and CSA) to effectively capture global contextual information from different feature spaces. Second, we integrate the GFL block with a multi-scale GCN-based LFA block to form a GLocal information extraction module, which highlights critical features of point clouds. Third, we stack a series of GLocal modules into a novel hierarchical encoder-decoder structure for multi-scale feature learning. Extensive experimental comparisons validate the effectiveness of 3DGTN in both classification and segmentation tasks.

Limitations. Despite its strong performance, 3DGTN has two main limitations: (1) The downsampling and neighborhood search operations in geometric space are time-consuming and cannot adaptively handle deep semantic features; (2) The global self-attention mechanism remains computationally and memory intensive, particularly for large-scale point cloud segmentation. To address these issues, we introduce more efficient Transformer frameworks in Chapters 4 and 5.

Chapter 4

Fast Point Transformer with Dynamic Token Aggregation[3]

4.1 Introduction

As discussed in Chapter 3, the hierarchical structure typically involves two key steps: point cloud sampling and grouping. Currently, most hierarchical point cloud processing methods use the FPS [89] algorithm, sampling points evenly across the geometric space. However, FPS is computationally expensive, making it a bottleneck for processing large-scale point clouds. The primary reason for its high time complexity is that FPS iteratively selects points that are farthest from the already selected set based on pairwise Euclidean distances. At each iteration, the algorithm computes the distances between the current set of sampled points and all remaining points in the input point cloud. This process involves multiple passes through the data and requires maintaining and updating a distance matrix. For a point cloud with N points, this results in a time complexity of $\mathcal{O}(N^2)$ for a complete sampling operation. Such quadratic scaling becomes increasingly prohibitive as the size of the input point cloud grows, especially in large-scale scenarios. Furthermore, FPS only focuses on the geometric properties of point clouds, ignoring their semantic features. This causes neural networks to de-emphasize some fine-level object parts with significant semantic information.

[3]The content of the chapter has been submitted to Expert Systems with Applications with a paper entitled "Efficient Point Transformer with Dynamic Token Aggregating for LiDAR Point Cloud Segmentation", under review.

Additionally, after downsampling, kNN and ball query [89] are widely used for point cloud grouping. However, such grouping methods are also strictly based on the geometric properties of points. In this situation, the local feature aggregation tends to be disturbed by semantic heterogeneity in local neighbourhoods, especially for points at the boundaries of adjacent parts. The aforementioned deficiencies are particularly pronounced when dealing with large-scale LiDAR datasets comprising millions of points, resulting in exceedingly high computational and memory costs. Recently, similar to superpixel in image processing, there have been many superpoint-based methods [49, 113, 99], which utilized the initial input feature (e.g. XYZ coordinates and RGB attributes) to preliminarily generate semantic homogeneous clusters. However, as a pre-processing step before the deep learning network, these methods fail to perform dynamic sampling and clustering for hierarchically extracted semantic features at different levels in the network, limiting their performance.

To address the aforementioned issues, I propose a novel and fast Transformer-based point cloud representation framework named DTA-Former, introducing the dynamic token selection, aggregation, and reconstruction methods to 3D Transformers. Firstly, a Learnable Token Sparsification (LTS) block is proposed for the adaptive and fast selection of key tokens. Compared with FPS, it not only considers both local and global semantic information for decision-making but also significantly reduces the computational cost. Secondly, instead of local feature aggregation by the max/average pooling operation, a Weighted Cross-Attention (WCA)-based Dynamic Token aggregation (DTA) block is designed for fast token aggregation, aggregating semantic information for each sparsified token at a global scale. It provides the model with strong aggregated features while mitigating information loss. Finally, I propose a new W-net architecture equipped with additional Iterative Token Reconstruction (ITR) blocks for point cloud segmentation. The ITR block is placed at the end of *each module* of the network, mapping the sparsified tokens back into the original token set by utilizing their semantic relationships. DTA-Former achieves a strong overall accuracy of 94.0% on the ModelNeto40 classification dataset, similar to 3DGTN in terms of algorithm accuracy. Towards the ShapeNet and MS-LiDAR datasets, it also obtains excellent results with 86.7% of mIoU and 95.4% of OA respectively, slightly better than 3DGTN. It is worth noting that DTA-Former is much faster than 3DGTN, with over 2.5× faster in terms of inference speed in large-scale MS-LiDAR data processing.

The key characteristics of DTA-Former can be summarized as follows:

(1) A novel Learnable Token Sparsification (LTS) block is proposed, which considers both local and global semantic information for fast and adaptive selection of key points. In terms of inference speed, LTS is 60× faster than FPS.

(2) The Dynamic Token Aggregation (DTA) block is proposed, which is realized by using

a novel Weighted Cross-Attention (WCA) mechanism for semantic feature aggregation, enlarging the effective receptive field while mitigating information loss.

(3) A new Iterative Token Reconstruction (ITR) block is proposed for the dense prediction of point clouds. It maps the fine-aggregated tokens into original tokens through the cross-attention map generated by WCA (named WCA-map). The accurate token relationships captured by the WCA-map ensure precise token reconstruction.

(4) Based on aforementioned blocks, the DTA-Former is proposed for fast point cloud learning and representation. A novel W-net architecture is designed for point cloud-dense prediction, which has been proven to be more suitable for Transformer-based feature learning than the classic U-net. Extensive experiments on classification, part segmentation, and MS-LiDAR semantic segmentation datasets demonstrate the superiority of the method in terms of both accuracy and speed.

4.2 Methodology

This section shows the details of DTA-Former. The overall framework is first presented, and then its key components are introduced: LTS, DTA blocks, as well as the ITR block tailored for fast dense prediction.

4.2.1 Overview

The overall pipelines of DTA-Former for point cloud classification and segmentation are shown in Figure 4.1. For the classification task, similar to 3DGTN, the input point cloud is firstly fed into a stem MLP block [90] for higher-dimension token generation. Secondly, it is fed into several modules hierarchically for token sparsification, aggregation, and GLocal feature learning. Specifically, each module consists of three blocks: a LTS block, a DTA block, and a GFL block. They are designed for fast downsampling and fully exploring long-range dependencies among tokens. Lastly, the extracted features by the aforementioned modules are fed into an MLP head consisting of a max-pooling layer, two fully connected layers, batch normalization, and RELU activation, for final prediction results. For the segmentation task, instead of using the common U-net-style encoder-decoder architecture of 3DGTN, a novel W-net-style architecture is designed for fast dense prediction. Compared with the U-net design [89], the W-net is more suitable for Transformer-based feature learning. Similar to the classification network, the segmentation network consists

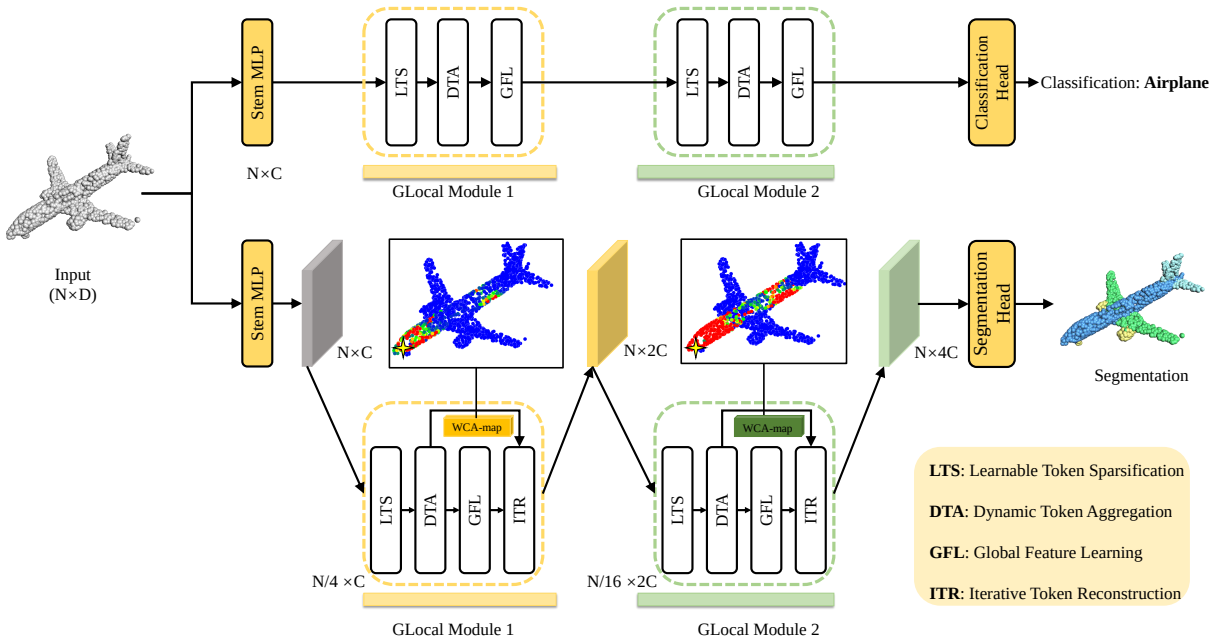


Figure 4.1: Pipelines of DTA-Former for classification (top) and segmentation (bottom).

of a series of modules containing LTS, DTA, and GFL blocks. Furthermore, each module contains an additional token reconstruction block for dense prediction, named ITR. It maps the sparsified tokens back into the original tokens by the WCA-map generated in the corresponding DTA block, gradually optimizing the semantic features of tokens and their semantic relationships during iterative reconstruction. The details of the ITR block and W-net architecture are elaborated in the following subsection. Lastly, an MLP head layer is used to get the final prediction for each point. Two scales of networks are designed for classification and segmentation tasks, namely DTA-Former-S and DTA-Former-L. The difference is that the feature dimension of the latter is twice that of the former at each module of the network.

4.2.2 Learnable Token Sparsification (LTS)

LTS hierarchically operates on all modules of both classification and segmentation networks. It drops redundant tokens dynamically according to learnable decision scores. As shown in Figure 4.2, given the input tokens $\mathbb{T} = \{T_i\}_{i=1}^{N} \in R^{N \times C}$, we first feed them into

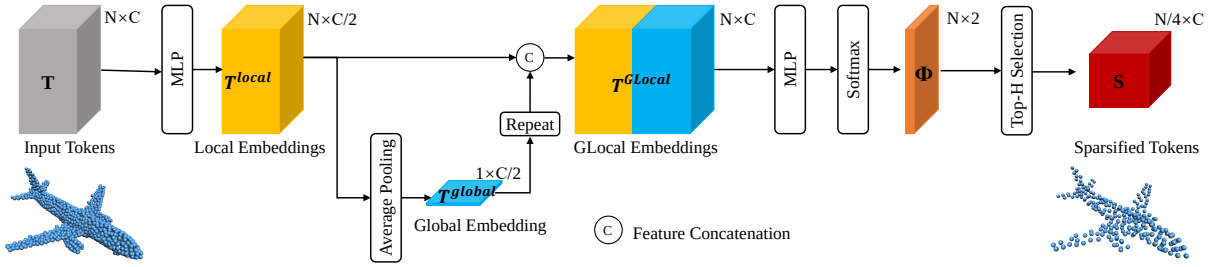


Figure 4.2: Illustration of LTS results on Module 1.

an MLP, getting the local embeddings:

$$\mathbb{T}^{local} = MLP(\mathbb{T}) \in R^{N\times C/2}, \tag{4.1}$$

where $\mathbb{T}^{local}$ is the local embedding of $\mathbb{T}$. Furthermore, a global feature for input tokens is computed by using average pooling, allowing the LTS block to analyze the importance of each token from a global perspective:

$$\mathbb{T}^{global} = Ave(MLP(\mathbb{T})) \in R^{1\times C/2}, \tag{4.2}$$

where $\mathbb{T}^{global}$ is the global embedding of $\mathbb{T}$. I then concatenate $\mathbb{T}^{local}$ and $\mathbb{T}^{global}$ to get the GLocal (global + local) embedding for $\mathbb{T}$, which contains both single token information and context information from the whole set:

$$\mathbb{T}^{GLocal} = \left[\mathbb{T}^{local}, \mathbb{T}^{global}\right] \in R^{N\times C}. \tag{4.3}$$

Given GLocal embedding set $\mathbb{T}^{GLocal}$, I compute the decision score set $\Phi = \{\varphi_i\}_{i=1}^{N} \in R^{N\times 2}$ for all tokens in $\mathbb{T}$:

$$\Phi = Softmax(MLP(\mathbb{T}^{GLocal})) \in R^{N\times 2}. \tag{4.4}$$

Φ_i has two feature channels representing keeping ($\varphi_{i,0}$) and dropping ($\varphi_{i,1}$) probabilities, respectively. Finally, Gumbel-Softmax [41] is employed to select the top-S tokens with the highest keeping probabilities, ensuring a differentiable process. Specifically, Gumbel-Softmax uses a reparameterization trick to approximate the discrete selection in a continuous manner, thereby addressing the non-differentiability of direct sampling operations. After the token sparsification process, a subset of $\mathbb{T}$ is obtained, which is denoted as $\mathbb{S} = \{S_i\}_{i=1}^{S} \in R^{S\times C}$. Thanks to the learnable decision scores, the token sparsification process is dynamically updated as the computation proceeds. The visualization results of sparsified tokens on different object categories are shown in Figure 4.3.

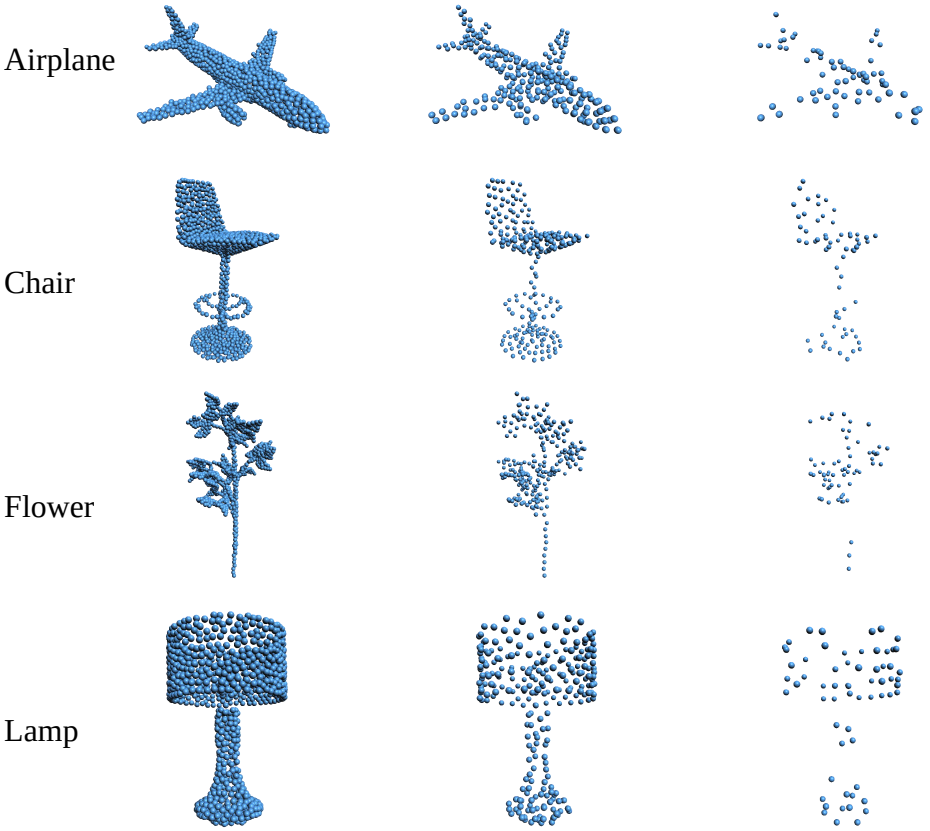


Figure 4.3: Visualization of sparsified tokens from LTS blocks.

4.2.3 Dynamic Token Aggregation (DTA)

The commonly used pooling approaches like max and average pooling are usually operated in local neighbourhoods for feature aggregation. This limits the receptive field and tends to cause information loss. Thanks to the great capability of long-range dependency modelling of the self-attention mechanism, I propose a DTA block with a new Weighted Cross-Attention (WCA) mechanism. It fully exploits the semantic relationships between $\mathbb{S}$ and $\mathbb{T}$ and achieves long-range feature aggregation adaptively for each sparsified token S_i, enlarging the effective receptive field while mitigating information loss. As shown in Figure 4.4, given the sparsified token set $\mathbb{S}$, I first generate a $Query$ matrix based on $\mathbb{S}$ and Key,

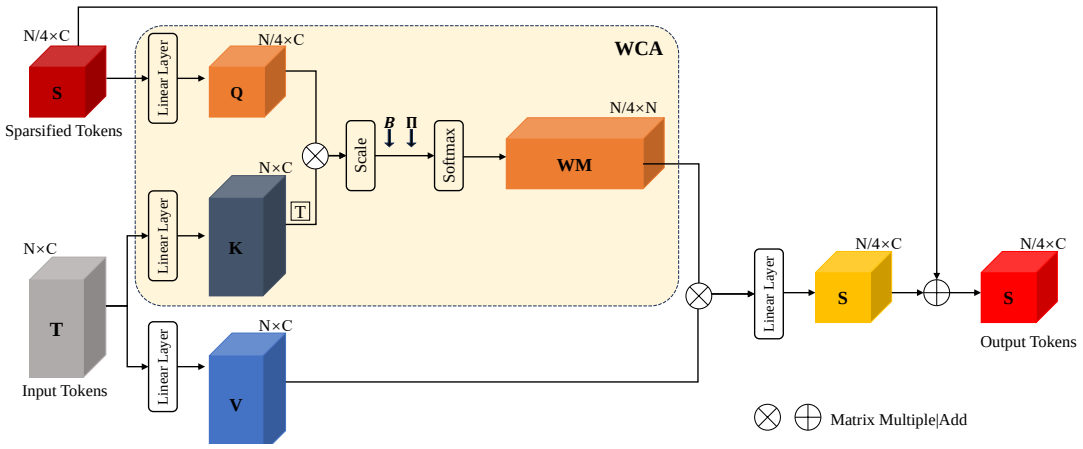


Figure 4.4: Illustration of the DTA block on Module 1.

$Value$ matrices based on $\mathbb{T}$:

$$\begin{aligned} Q &= \mathbb{S} \times W_Q, \\ K &= \mathbb{T} \times W_K, \\ V &= \mathbb{T} \times W_V, \end{aligned} \tag{4.5}$$

where Q, K, V represent $Query$, Key, $Value$ matrices, W_Q, W_K, W_V are learnable weight matrices. Secondly, the WCA-map WM is calculated as:

$$WM = Softmax(\Pi \odot (\frac{Q \times K^T}{\sqrt{C}} + B)) \in R^{S \times N}, \tag{4.6}$$

where B is a learnable position encoding matrix defined by [160], $\odot$ represents the Hadamard operation, i.e., an element-wise multiplication between two matrices (or vectors), $\Pi = \{\varphi_{i,0}\}_{i=1}^{N} \in \mathbb{R}^{N \times 1}$ is a vector of decision scores in Section 4.2.2, whose components are used as weights for input tokens. In detail, since the shape of Π is $N \times 1$, which is inconsistent with QK^T, I first reshape Π to $1 \times N$, followed by expanding it to $S \times N$ by row repetition. Π essentially represents the importance of each point to network decision-making. The combination of Π and the attention map allows the model to focus more on key tokens with high scores, which is beneficial to enhance the representation ability of the network. The ablation study in Section 4.3.5 also demonstrates the superiority of WCA. Finally, WM is taken as a weight matrix for global token aggregation, and multiplied by V to generate the final pooled sparsified token set $\mathbb{S}$ with the size of $S \times D$. Compared with local

grouping and pooling operations, DTA exploits the long-range context information better for feature aggregation, while being faster due to avoiding neighbourhood construction.

4.2.4 Iterative Token Reconstruction (ITR)

ITR is unique to the segmentation network, tailored to dense prediction. Given the enhanced token set $\mathbb{S}$, ITR maps it back into the original token set $\mathbb{T}$ by the WCA-map WM in Eq. 4.6, followed by a residual connection:

$$\mathbb{T} = Softmax(WM^T) \times \mathbb{S} + \mathbb{T} \in R^{N \times C}. \tag{4.7}$$

As shown in Figure 4.1, the ITR block is placed in every module of the segmentation network, forming a W-net architecture. Consequently, input tokens can be dynamically reconstructed according to the WCA-map stored in DTA, gradually refining their semantic features and inter-token relationships during iterative reconstruction. Figure 4.5 provides a concise illustration of how the WCA-map interacts with the ITR block.

We also applied the ITR block to a U-net architecture (Section 4.3.5), but its performance dropped significantly compared to the W-net. In the U-net, WCA-maps generated by the encoder are less effective for tokens that have undergone deep feature learning in the decoder. By contrast, the proposed segmentation network offers immediate feedback of WCA-maps to sparsified tokens, enabling more effective reconstruction.

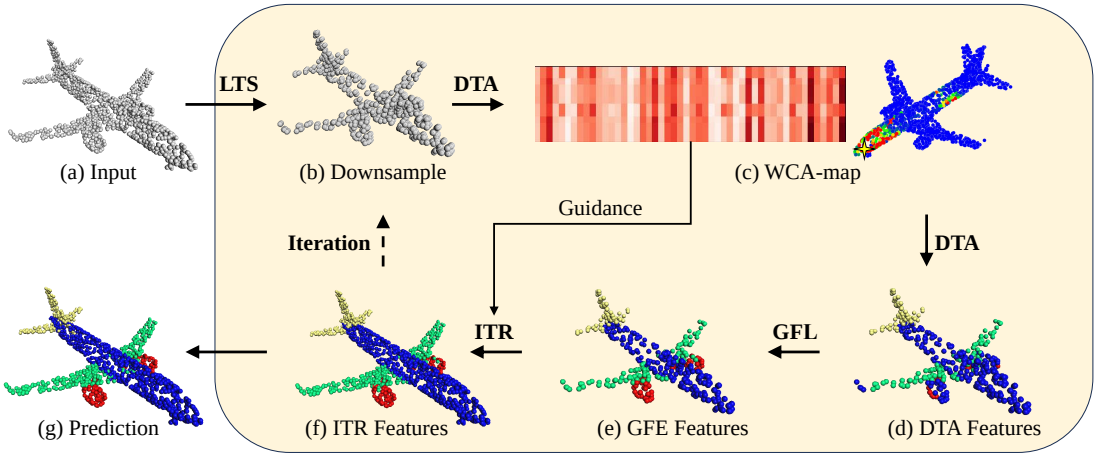


Figure 4.5: Brief illustration of the segmentation network.

4.3 Experimental Results and Discussion

This section first presents the implementation details of DTA-Former, including hardware configuration, training strategy, and hyperparameter settings. Secondly, similar to 3DGTN, I present the performance of the method on the classification dataset (ModelNet40 [135]), part segmentation dataset (ShapeNet [146]), and remote sensing semantic segmentation (Airborne MS-LiDAR dataset [161]). Lastly, I conducted ablation studies to verify the effectiveness of key components in DTA-Former.

4.3.1 Implementation Details

DTA-Former was implemented with PyTorch. It was trained with the SGD Optimizer, with a momentum of 0.9 and weight decay of 0.0001. The initial learning rate was set to 0.01 for classification and 0.1 for segmentation, with a cosine annealing schedule to adjust the learning rate at every epoch. The network was trained for 300 epochs with a batch size of 16 for the classification task, and 200 epochs with a batch size of 8 for the segmentation task. To make a fair comparison, the experiments on model latency in this thesis are all tested on an NVIDIA GeForce RTX 2080 GPU.

4.3.2 Point Cloud Classification

Datasets and Metrics. To make a fair comparison, DTA-Former was also tested on the ModelNet40 dataset for the classification task, following the same data preprocessing as 3DGTN. Similarly, metrics of OA and latency are used for performance evaluation.

Performance Comparison. As shown in Table 4.1, with the same input size, both DTA-Former-S and DTA-Former-L achieve excellent classification results as measured by OA, competitive with 3DGTN and outperforming other prior SOTA Transformer methods such as PointTransfromer [160], PatchFromer [151], and PointConT [64]. This demonstrates the excellent performance of DTA-Former in point cloud classification. In terms of algorithm speed, DTA-Former-S has the lowest latency (32.4ms) among all involved Transformer-based methods, over 2× faster than 3DGTN and 16× faster than PointTransformer [160]. It also exceeds prior efficient 3D Transformers such as PatchFromer [151].

Table 4.1: Quantitative comparison of DTA-Former on the ModelNet40 dataset.

Methods	Input Size	OA (%)	Latency (ms)
Other Learning-based Methods			
PointNet++[89]	1024	91.9	35.3
PointCNN[57]	1024	92.2	221.2
DGCNN[127]	1024	92.2	85.8
DeepGCN[51]	1024	93.6	81.2
PointNeXt[90]	1024	93.7	32.7
RepSurf[95]	1024	**94.0**	99.5
Point-PN[155]	1024	93.8	45.2
APES[131]	1024	93.8	**14.5**
Transformer-based Methods			
PointASNL[144]	1024	93.2	923.6
PCT[30]	1024	93.2	92.4
PT[160]	1024	93.7	530.2
GBNet[92]	1024	93.8	103.7
PatchFormer[151]	1024	93.5	34.3
PointConT[64]	1024	93.5	79.1
3DGTN [70]	1024	**94.0**	66.7
DTA-Former-S	1024	93.8	**32.4**
DTA-Former-L	1024	**94.0**	52.5

4.3.3 Point Cloud Part Segmentation

Datasets and Metrics. To make a fair comparison, DTA-Former was also tested on the ShapeNet dataset for the part segmentation task, following the same data preprocessing as 3DGTN. Similarly, the metrics of mIoU and latency are used for performance evaluation.

Performance Comparison. Table 4.2 shows the comparison results of DTA-Former and existing SOTA methods on the ShapeNet dataset. As measured by mIoU, DTA-Former-L achieves the best results (86.7%) compared with the prior SOTA methods such as 3DGTN, PT [160], Stratified Transformer [47], and APES [131]. With similar accuracy, DTA-Former-S achieves the lowest latency (37.4ms) compared to efficient Transformers represented by PatchFormer [151]. Specifically, it is 3.6× faster than 3DGTN, 14.9× faster than PT [160], 47.9× faster than Stratified Transformer [47], and 1.2× faster than efficient PatchFormer [151]. These demonstrate the excellent performance of DTA-Former in part segmentation. The visualization of part segmentation results are shown in Figure

4.6, as well as the WCA-maps from the DTA block. From the results, the segmentation predictions are faithful to the ground truth. Moreover, the WCA-map visualization reveals a strong correlation between attention weights and point semantics, which illustrates the effectiveness of the proposed DTA in semantic information aggregation.

Table 4.2: Part segmentation results of DTA-Former on the ShapeNet dataset.

Methods	Input Size	mIoU (%)	Latency (ms)
Other Learning-based Methods			
PointNet++(MSG)[89]	2048	85.1	47.7
PointCNN[57]	2048	86.1	252.1
DGCNN[127]	2048	85.2	96.2
DeepGCN[51]	2048	86.3	105.6
RepSurf[95]	2048	86.4	117.1
Point-PN[155]	2048	**86.6**	58.9
APES[131]	2048	85.8	**24.8**
Transformer-based Methods			
PointASNL[144]	2048	85.4	1023.2
PCT[30]	2048	86.4	101.1
PT[160]	2048	86.6	560.2
GBNet[92]	2048	85.9	198.4
Stratified Transformer[47]	2048	86.6	1792.6
PatchFormer[151]	2048	86.5	45.8
Li et al.[58]	2048	86.0	-
Hassan et al.[38]	2048	86.3	-
3DGTN [70]	2048	86.6	136.8
DTA-Former-S	2048	86.5	**37.4**
DTA-Former-L	2048	**86.7**	59.3

4.3.4 Airborne MS-LiDAR Segmentation

Datasets and Metrics. To make a fair comparison, DTA-Former was also tested on the airborne MS-LiDAR dataset for the semantic segmentation task, following the same data preprocessing as 3DGTN. Similarly, the average F_1 score [137], mIoU, OA, and latency are used for performance evaluation.

Performance Comparison. The confusion matrix of the MS-LiDAR data segmentation results is shown in Table 4.3, where the numbers in the last three rows represent the Precision, Recall, and F_1 score for each category. As we can see, all the F_1 scores of

all categories except soil are over 85%, where the F_1 scores of Building, Grass, and Tree exceed 90%. Most of the misclassified points are found in areas of soil and roads. This is because the similar geometric characteristics of the soil and roads tend to confuse the network. The comparison results are shown in Table 4.4[4]. Compared with prior SOTA methods, DTA-Former-L achieves competitive results in terms of OA (95.4%), average F_1 score (88.3%), and mIoU (81.4%) with 3DGTN, where OA is slightly better than 3DGTN. As measured by latency, DTA-Former-L and and DTA-Former-S are 2.75× and 5× faster than 3DGTN respectively.

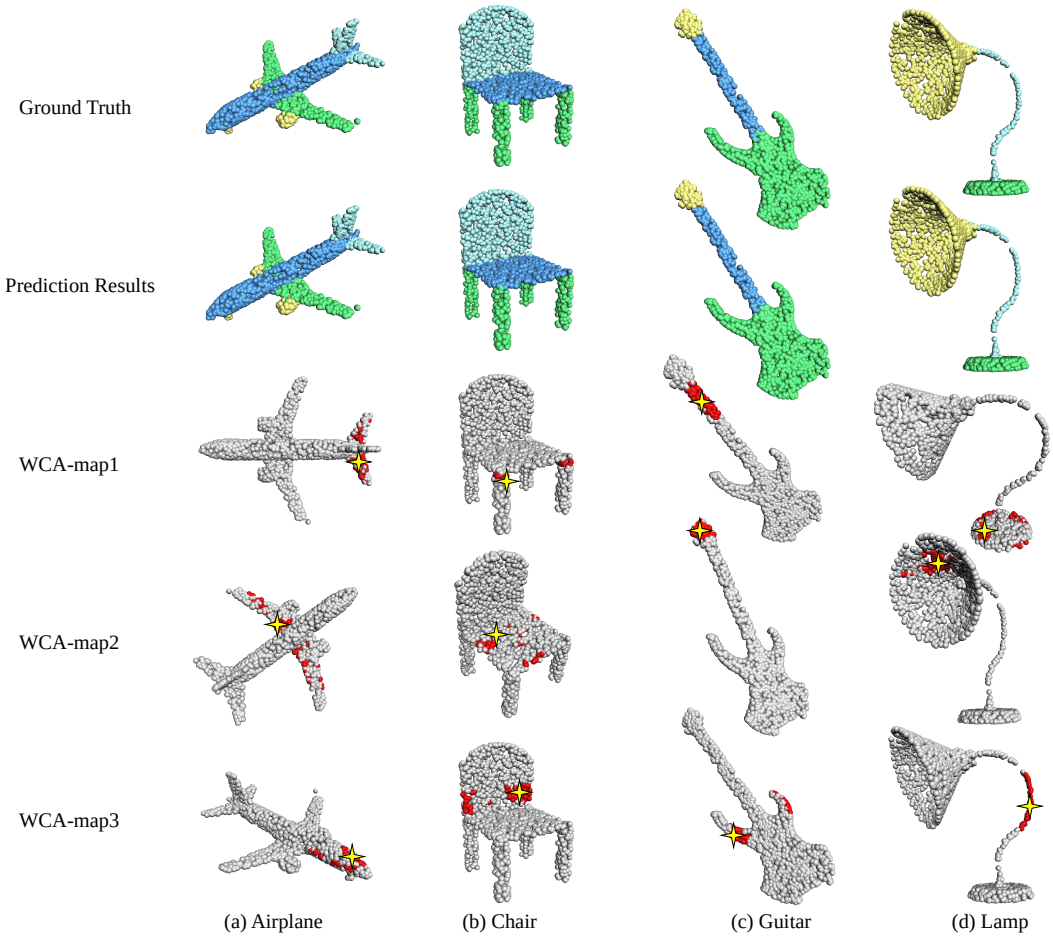


Figure 4.6: Illustration of part segmentation and WCA-map results on ShapeNet.

[4]Methods marked with * were not originally tested on this dataset in their respective papers. The reported results were obtained by running the official code provided by the authors.

Table 4.3: Confusion matrix (%) of DTA-Former-L on the airborne MS-LiDAR dataset.

Categories		True Label					
		Road	Building	Grass	Tree	Soil	Powerline
Prediction Label	Road	82.5	1.5	0.0	0.0	5.3	0.0
	Building	14.1	95.7	0.7	0.3	38.6	0.0
	Grass	0.4	1.2	98.8	2.0	0.1	13.4
	Tree	0.1	0.2	0.4	97.7	0.2	1.2
	Soil	2.8	1.2	0.0	0.0	56.0	0.0
	Powerline	0.0	0.0	0.0	0.0	0.0	85.2
Precision		93.1	89.1	99.0	94.1	77.4	94.3
Recall		82.5	95.7	98.8	97.7	56.0	85.2
F_1		87.5	92.4	98.9	95.9	64.9	90.1

The visualization results of comparison methods on testing area-11, 12, and 13 are shown in Figures 4.7, 4.8 and 4.9. These results demonstrate the excellent performance of DTA-Former in MS-LiDAR point cloud segmentation, exceeding previous SOTA methods. Prediction details of several samples from the testing area-11 are shown in Figure 4.10, with the LTS results in different network modules.

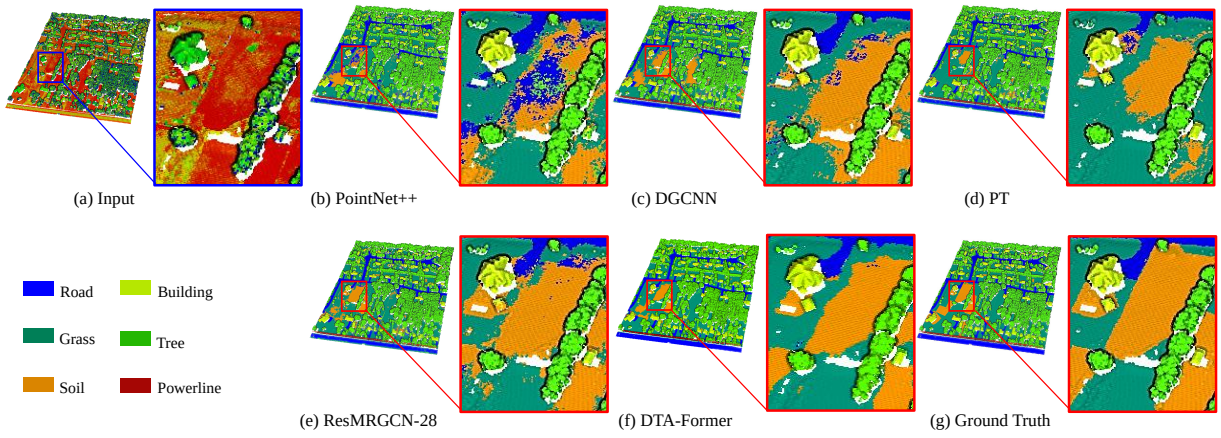


Figure 4.7: Comparison results from different methods on the testing area-11 in the airborne MS-LiDAR dataset.

Table 4.4: Quantitative comparison of DTA-Former on the MS-LiDAR dataset.

Methods	Average F_1 score (%)	mIoU (%)	OA (%)	Latency (ms)
PointNet++ [89]	72.1	58.6	90.1	322.6
DGCNN [127]	71.6	51.0	91.4	86.2
RSCNN [66]	73.9	56.1	91.0	158.7
GACNet [122]	67.7	51.0	90.0	277.8
AGConv [164]	76.9	71.2	93.3	312.5
SE-PointNet++ [43]	75.9	60.2	91.2	-
FR-GCNet [161]	78.6	65.8	93.6	-
PointTransformer2* [160]	80.5	73.6	93.1	285.7
PPT-Net* [40]	80.1	73.6	92.7	43.3
PatchFormer* [151]	82.4	77.8	93.1	62.9
Xiao et al. [137]	83.3	79.3	94.0	-
ResMRGCN-28* [52]	81.1	74.0	93.3	45.7
GCNAS [154]	88.1	**82.3**	95.2	-
3DGTN [70]	**88.6**	82.1	95.2	217.4
DTA-Former-S	87.8	80.9	95.1	**43.1**
DTA-Former-L	88.3	81.4	**95.4**	78.9

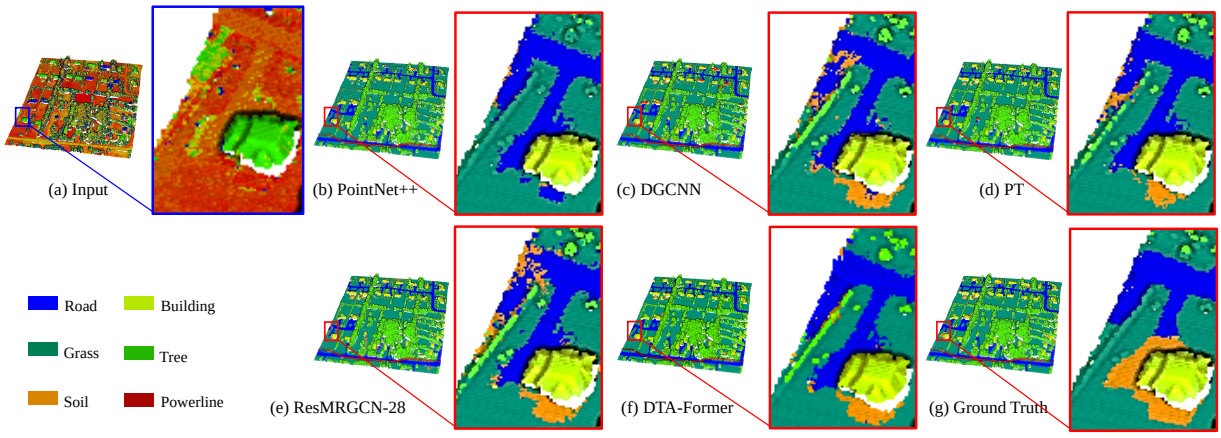


Figure 4.8: Comparison results from different methods on the testing area-12 in the airborne MS-LiDAR dataset.

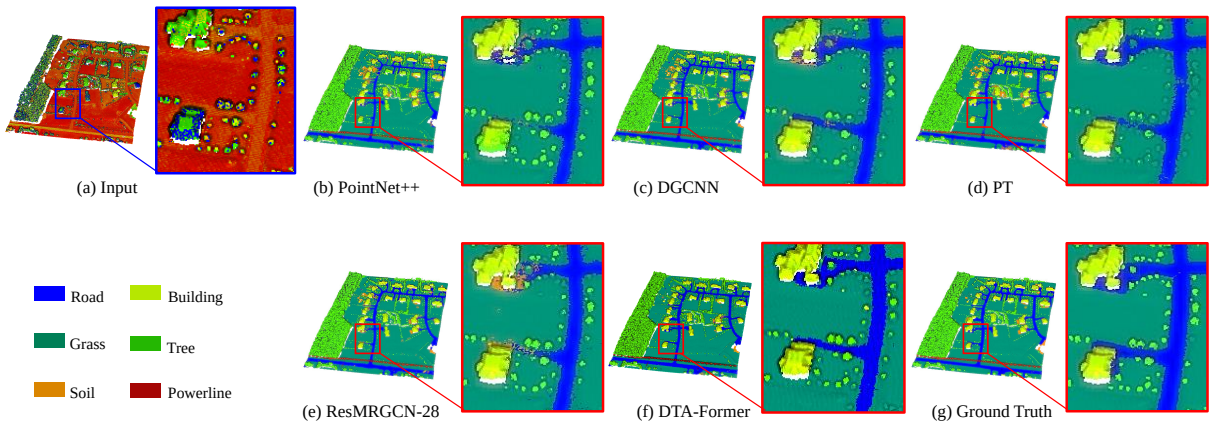


Figure 4.9: Comparison results from different methods on the testing area-13 in the airborne MS-LiDAR dataset.

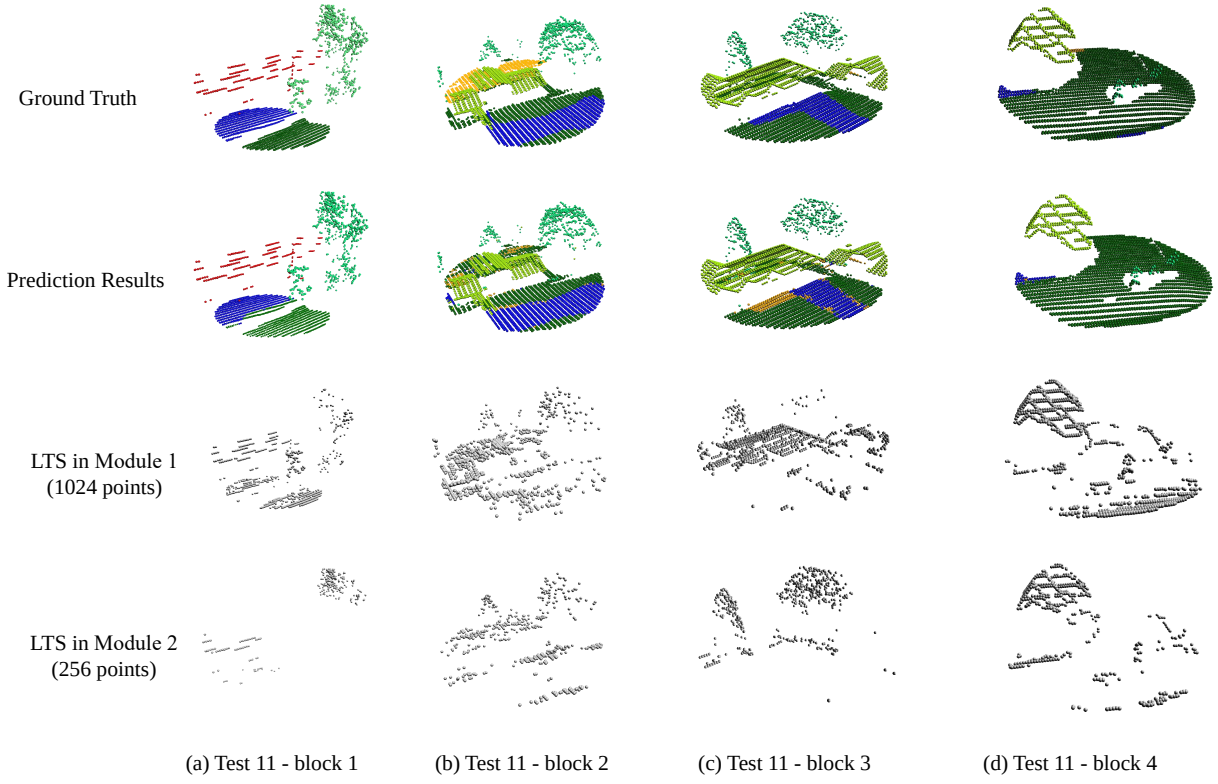


Figure 4.10: Visualization results of samples from the testing area-11.

4.3.5 Ablation Study

Taking DTA-Former-L as the baseline, I conducted a series of ablation studies on the airborne MS-LiDAR dataset to verify the effectiveness of key blocks in DTA-Former.

Table 4.5: Ablation results for DTA-Former-L on the airborne MS-LiDAR dataset.

Ablation		Average F_1 (%)	mIoU (%)	OA (%)	Latency (ms)
LTS	→FPS	87.4	79.3	95.1	227.4
	→Random Sampling	80.2	74.9	90.7	78.8
DTA	–	80.5	75.7	91.9	**68.8**
	→kNN + MLP	81.7	77.4	91.1	88.3
	WCA→VCA	84.3	79.1	94.0	78.9
ITR	→Trilinear interpolation	83.1	78.4	92.8	82.7
	→Nearest interpolation	82.5	78.2	92.7	78.8
	Equipped in U-net	81.5	75.3	90.8	73.5
DTA-Former-L (Baseline)		**88.3**	**81.4**	**95.4**	78.9

Learnable Token Sparsification. The effectiveness of the LTS block was first investigated by compared to the commonly used FPS, and random sampling approaches. As shown in Table 4.5 Row 2-3, DTA-Former is much faster than that with FPS in terms of latency. Although LTS is slightly lower than random sampling in terms of inference speed, it outperforms random sampling by 8.1 absolute percentage points in terms of average F_1 score. This is because random sampling is very sensitive to point cloud density, which leads to its poor performance in real-world LiDAR data processing.

Dynamic Token aggregation. A series of ablation experiments were conducted to verify the effectiveness of DTA. I first removed the DTA block from the network, which means the output sparsified tokens are directly fed into the GFL block in each module. Accordingly, the token reconstruction process was realized by the common nearest interpolation method due to the lack of the WCA-map WM. As shown in Table 4.5 Row 4, the performance of the model without DTA drops significantly, where OA drops by 3.5 percentage points, mIoU drops by 5.7 percentage points, and average F_1 score drops by 7.8 percentage points. Secondly, I replaced DTA with a common local feature aggregation method, kNN + MLP. From Table 4.5 Row 5, there is a drop in both the accuracy and latency of the method. These results demonstrate the superiority of DTA. Finally, I replaced the WCA mechanism with a Vanilla Cross-Attention (VCA) one. As shown in Table 4.5 Row 6, we can observe a 1.4, 2.3, and 4.0 drop in OA, mIoU, and average F_1 score respectively. This suggests that the WCA mechanism contributes to model performance.

Iterative Token Reconstruction. Table 4.5 Row 7-9 shows the ablation studies of ITR. I first replaced the ITR block with the common trilinear and nearest neighbour interpolations, respectively. From the results, we can see that the proposed ITR block outperforms the aforementioned two kinds of interpolation methods, demonstrating the benefits of the accurate token-wise semantic relationship captured by DTA. Instead of the W-net architecture, I also explored the segmentation performance of the proposed ITR in the U-net architecture, while keeping other modules unchanged. As shown in Table 4.5 Row 9, the performance of the U-net with ITR drops significantly compared with the W-net architecture. In the U-net architecture, the feature maps in the encoder need to undergo further feature extraction and reconstruction to obtain the corresponding feature maps in the decoder. Therefore, the feature relationships in the decoder cannot be accurately described by the corresponding WCA-maps established by the DTA block in the encoder. In contrast, the W-net segmentation network allows the WCA-maps in the DTA block to be promptly fed back to the sparsified tokens in the same module for reconstruction. This ensures the timeliness of feature interaction and reconstruction.

4.4 Chapter Summary

This Chapter proposes a fast point Transformer network, named DTA-Former for point cloud representation, consisting of several novel blocks such as LTS, DTA, and ITR. For the classification task, DTA-Former consists of two main modules, where each module is composed in a sequence of LTS, DTA, and GFL blocks. Specifically, the LTS block calculates a decision score for each input token by considering its both local and global information, followed by token sparsification with a differentiable top-S selection method. The proposed LTS significantly outperforms FPS, 60$\times$ faster than FPS as measured by model latency. The sparsified tokens are then fed into the DTA block for feature aggregation at a global scale. In the DTA block, a novel WCA is proposed to fully explore the semantic relationship between the sparsified token set and the original token set. Therefore, instead of in a fixed local region, DTA allows each sparsified token to dynamically search its semantic-homogeneity neighbourhood in a global view, generating an adaptive receptive field while mitigating information loss which is common in traditional pooling operations. Following the DTA block, a dual-attention Transformer-based GFL block is still employed to enhance the ability of the model in long-range dependency modelling. For the segmentation task, instead of the common U-net design, a novel W-net architecture is proposed for dense prediction tasks, which is achieved by using ITR. In addition to the aforementioned three blocks, the ITR block is placed at the end of each module of the

segmentation network, mapping the sparsified tokens back into the original token set by the WCA-map generated in the DTA block. From the ablation experiments, compared with the U-net [89], the proposed W-net is more suitable for Transformer-based feature learning. Extensive experiments on the ModeleNet40 classification [135], ShapeNet part segmentation[146] and airborne MS-LiDAR semantic segmentation datasets [161] demonstrate that DTA-Former significantly outperforms 3DGTN in terms of model latency, while keeping competitive accuracy.

Limitations. There are two main limitations in DTA-Former. One is that although the introduction of LTS and DTA blocks reduces the time complexity of the model (i.e., the inference time of DTA-Former is much less than 3DGTN), they also increase the space complexity of the model (i.e., DTA-Former consumes more GPU memory than 3DGTN (3296 MB vs. 2666 MB)). Another one is that the LTS block is essentially a hard sampling strategy, which still tends to cause information loss when discarding other low-score points. Besides, the feature clustering process based on the combination of LTS and DTA still has room for improvement. Therefore, I further optimize the DTA-Former framework, reducing both time complexity and space complexity by introducing 'Learnable Supertokens' to Transformers (Chapter 5).

Chapter 5

3D Learnable Supertoken Transformer[5]

5.1 Introduction

Most recently, integrating superpoints [49] with 3D Transformer techniques [113, 99] has achieved tremendous progress in terms of both efficiency and accuracy improvement of point cloud segmentation. The superpoint clustering strategy reduces the number of input tokens to the Transformer network significantly and provides relatively accurate over-segmentation results, which facilitates the subsequent feature learning and understanding. Inspired by these existing works, I found that the introduction of superpoints can replace the LTS block in DTA-Former, further reducing the space complexity of the model. Besides, since the number of superpoints could be much less than that of sampling points in DTA-Former, the computational complexity could be further reduced.

Despite the great improvement achieved by superpoint Transformers, the limited initial point features involved in superpoint construction and its time-consuming computations tend to hinder further improvement of model performance. To address this issue, this chapter proposes a Learnable Supertoken Transformer framework, named 3DLST, which is the first work to introduce learnable supertoken and dynamic optimization in the 3D Transformer paradigm. Compared with the aforementioned static superpoint clustering method, the proposed Dynamic Supertoken Optimization (DSO) block can optimize the

[5]The content of the chapter has been submitted to International Journal of Applied Earth Observation and Geoinformation, with a paper entitled "3DLST: 3D Learnable Supertoken Transformer for LiDAR Point Cloud Scene Segmentation", under revision.

supertokens dynamically based on multi-level deep features. Therefore, the learnable supertokens are beneficial to the semantic homogeneity-aware token clustering process, which is more flexible and efficient.

In summary, the main characteristics of 3DLST are as follows:

(1) The first Dynamic Supertoken Optimization (DSO) block is proposed in the 3D Transformer paradigm for efficient token clustering and aggregating. The proposed learnable supertokens serve the semantic homogeneity-aware token clustering process directly, outperforming traditional superpoint methods.

(2) An efficient Cross-Attention-guided Upsampling (CAU) block tailored to the supertoken clustering is proposed for token reconstruction. It fully exploits the long-range context dependency modelling capability of the Transformer, achieving the upsampling process by rich semantic similarity information among tokens.

(3) A 3D Learnable Supertoken Transformer (3DLST) network is proposed, equipped with a novel W-net architecture tailored to 3D Transformer-based point cloud segmentation. By integrating novel DSO and CAU blocks into the W-net architecture, 3DLST achieves the SOTA performance in terms of both accuracy and efficiency on various LiDAR datasets.

5.2 Methodology

This section details the 3DLST framework. Firstly, the overall design of 3DLST is introduced. Secondly, the key components in 3DLST are presented: Dynamic Supertoken Optimization (DSO) and Cross-attention-guided Upsampling (CAU) blocks.

5.2.1 Overview

The overall framework of 3DLST is shown in Figure 5.1. It first applies a powerful and efficient feature extractor achieved by Point-Voxel CNN (PVCNN) [68] to achieve an initial local embedding for input tokens. The embedded tokens are denoted as $\mathbb{T} = \{T_i\}_{i \in N} \in R^{N \times C}$. Secondly, a novel supertoken generation and optimization strategy (i.e., DSO) is proposed for token clustering and aggregation, replacing the LTS and DTA blocks in DTA-Former. Specifically, it starts from randomly initialized learnable embeddings, defined as initial supertoken. Then a hard-assignment cross-attention mechanism is applied for dynamic token

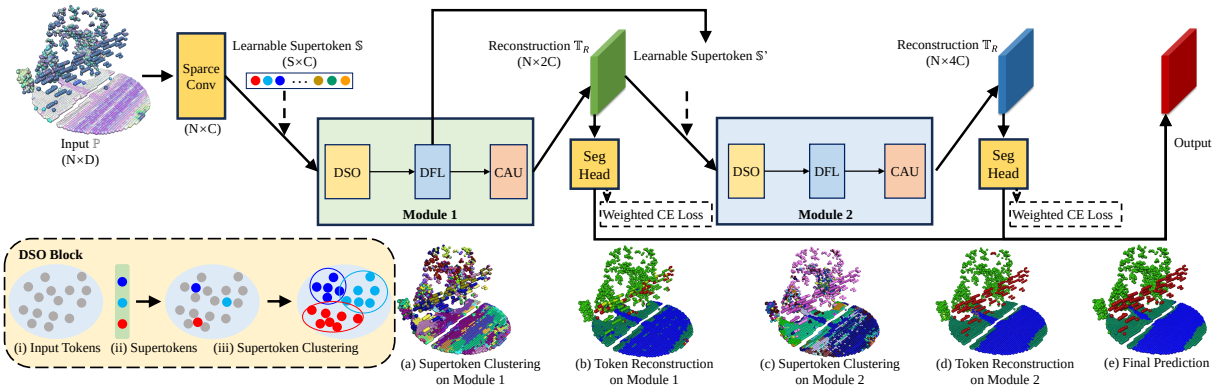


Figure 5.1: Architecture of 3DLST.

clustering based on the initial supertoken. After that, an updated supertoken set could be obtained by an average pooling operated on each cluster, followed by further enhancement via the GFL block. Since the number of supertokens is much less than that of sampling points in DTA-Forner, the computational complexity of the GFL block could be reduced. Furthermore, to adapt to the newly proposed DSO block, an efficient upsampling block (i.e., CAU) is developed for token reconstruction, which is similar to the ITR block in DTA-Former. As shown in Figure 5.1, Module 2 has a similar structure to Module 1, where the enhanced supertokens from Module 1 are further updated in Module 2. Module 2 outputs reconstructed tokens with higher-dimension features than Module 1.

After the aforementioned two modules, two sets of reconstructed tokens with different feature dimensions can be obtained. Finally, an MLP-based segmentation head is applied to each token set and a multiloss strategy is adopted for network training. During testing, the sum of the predicted probabilities of the two token sets is taken as the final result.

5.2.2 Dynamic Supertoken Optimization (DSO)

Taking the embedded tokens from the feature extractor as input, the DSO block defines a learnable supertoken set as clustering centers, denoted as $\mathbb{S} = \{S_i\}_{i \in S} \in R^{S \times C}$. $\mathbb{S}$ is randomly initialized and gradually optimized during network training. Compared with the traditional superpoint constructing approaches which are usually based on limited initial attributes of the input points, DSO serves the semantic homogeneity-aware token clustering process directly, which is more flexible and efficient. Compared with DTA-Former, on the one hand, the definition of learnable supertokens replaces the point sampling process of the

LTS block, which not only reduces the computational complexity and memory consumption of the model. Besides, since the number of supertokens is much less than that of sampling points in DTA-Former, the model efficiency is further improved. On the other hand, DSO avoids the information loss caused by discarding low-score points in the LTS block, leading to a further improvement in model accuracy.

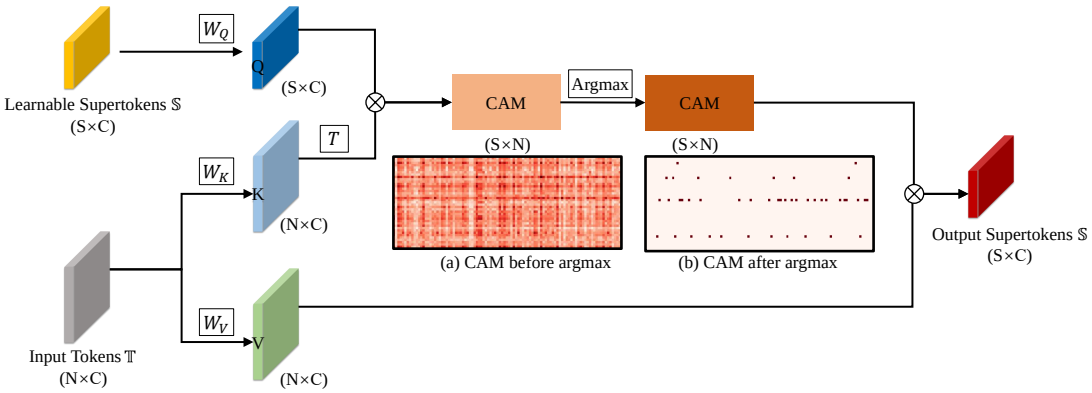


Figure 5.2: Illustration of the cross-attention calculation.

As shown in Figure 5.2, given the embedded tokens $\mathbb{T}$ and the supertoken set $\mathbb{S}$, a cross-attention is performed to explore the semantic similarity between $\mathbb{T}$ and $\mathbb{S}$. Firstly, The $Query$, Key, and $Value$ matrices are computed as below:

$$\begin{aligned} Q &= \mathbb{S} \times W_Q, \\ K &= \mathbb{T} \times W_K, \\ V &= \mathbb{T} \times W_V, \end{aligned} \tag{5.1}$$

where Q, K, and V represent $Query$, Key, and $Value$ matrices, and W_Q, W_K, and W_V are learnable weight matrices. After that, a Cross-Attention Map (CAM) is calculated to reflect the feature similarity between Q and K:

$$CAM = \arg\max_N \left(\frac{Q \times K^T}{\sqrt{C}}\right). \tag{5.2}$$

From Eq. 5.2, the $argmax$ operation performs a hard assignment from the K set to the Q set, which is inspired by the K-means clustering strategy [147]. Ablation experiments (Section 5.3.4) show the $argmax$ operation achieves better results than $softmax$ in 3DLST. By the hard assignment, the semantic-homogeneity cluster for each supertoken can be constructed. Finally, the supertoken set $\mathbb{S}$ could be updated by averaging tokens in the same cluster.

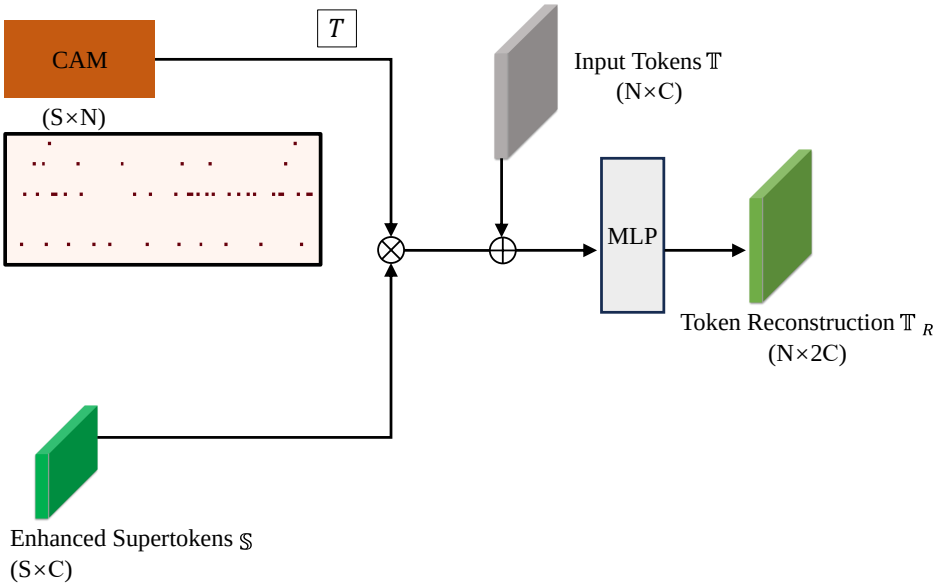


Figure 5.3: Illustration of the CAU block.

5.2.3 Cross-Attention-guided Upsampling (CAU)

Given the enhanced supertoken set $\mathbb{S}$ by the DFL block, a simple yet effective upsampling block is proposed for token reconstruction. As shown in Figure 5.3, since the clustering relationship has been recorded in the CAM, the token reconstruction process can be formulated as below.

$$\mathbb{T}_R = MLP(\mathbb{T} + CAM^T \times \mathbb{S}) \in R^{N \times 2C}. \tag{5.3}$$

The CAU block is placed as the last block in each module of 3DLST, resulting in the W-net architecture. As such, the input tokens could be optimized dynamically during network training. Compared with the symmetric U-net design [89], the proposed W-net architecture allows the CAM to provide prompt guidance for token reconstruction, resulting in better prediction results. Fig .5.4 shows the prediction results of token reconstruction in each module. From the results, the prediction result from Module 2 is generally better than that from Module 1, which confirms the dynamic optimization of token reconstruction in 3DLST. Additionally, prediction results from both Module 1 and 2 get better and better as the network is trained, which illustrates the strong feature learning capabilities of 3DLST.

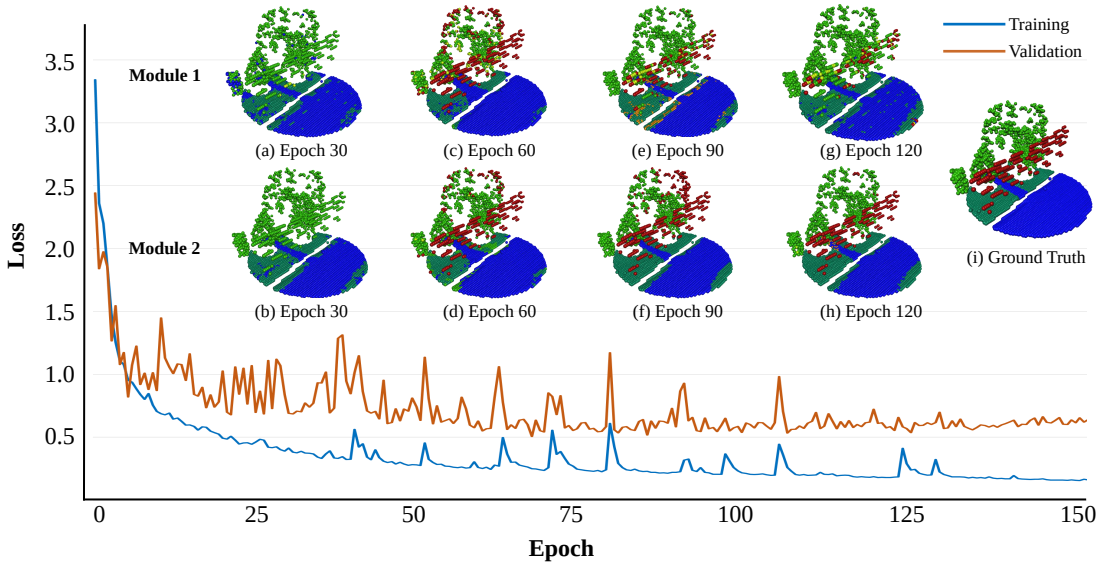


Figure 5.4: The training and validation loss curves during network training.

5.3 Experimental Results and Discussion

This section first introduces the implementation details of 3DLST, including hardware configuration and hyperparameter settings for different datasets (airborne MS-LiDAR datasets [161] and DALES [119]). Secondly, the performance evaluation and comparison of 3DLST is presented. Finally, ablation studies on the main components of the method are presented.

5.3.1 Implementation Details

3DLST was implemented in PyTorch. It was trained with 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 for learning rate adjusting. The batch size was set to 16 for the airborne MS-LiDAR dataset, and 8 for the DALES datasets, according to different input sizes. For each dataset, the network was trained for 250 epochs, with the weighted Cross-Entropy (CE) loss function. To make a fair comparison, the experiments on model efficiency in this thesis are all tested on an NVIDIA GeForce RTX 2080 GPU.

5.3.2 Airborne MS-LiDAR Dataset

Performance Comparison. A detailed performance evaluation of 3DLST on the airborne MS-LiDAR dataset is shown in Table 5.1, presented as a confusion matrix. The Precision, Recall, and F_1 score for each category are presented in the last three rows of the table. From the results, 3DLST achieves F_1 scores of over 85% for all categories except soil, where the F_1 scores of grass and tree exceed 98%.

Table 5.1: Confusion matrix (%) of 3DLST on the airborne MS-LiDAR dataset.

Categories		**True Label**					
		Road	Building	Grass	Tree	Soil	Powerline
Prediction Label	Road	89.0	2.2	0.0	0.0	17.3	0.0
	Building	9.6	95.6	0.5	0.1	28.5	0.0
	Grass	0.3	1.3	99.3	0.6	0.0	15.1
	Tree	0.0	0.2	0.2	99.4	0.0	1.0
	Soil	1.0	0.7	0.0	0.0	54.2	0.0
	Powerline	0.0	0.0	0.0	0.0	0.0	83.8
Precision		87.8	92.0	99.1	97.0	87.2	96.2
Recall		89.0	95.6	99.3	99.4	54.2	83.8
F_1		88.4	93.8	99.2	98.2	66.8	89.6

Moreover, comparison results shown in Table 5.2[6] demonstrate that 3DLST surpasses the previous SOTA methods including 3DGTN and DTA-Former in terms of both accuracy and efficiency. Specifically, 3DLST is over 6.5× faster than DTA-Former, over 5× faster than PatchFormer [151], and about 4× faster than ResMRGCN-28 [52]. The visualization of the segmentation result comparison is shown in Figures 5.5, 5.6, and 5.7, showing the excellent segmentation performance of 3DLST. Additionally, the clustering results of supertokens in 3DLST are shown in Figure 5.8. From the results, we can see that the clustering results of 3DLST are visually scattered. This is because the proposed supertokens focus on splitting the input samples into a series of semantic-homogeneity groups, instead of geometry-based clustering. Overall, both comparison and visualization results demonstrate the superiority of 3DLST in MS-LiDAR point cloud scene segmentation.

[6]Methods marked with * were not originally tested on this dataset in their respective papers. The reported results were obtained by running the official code provided by the authors.

Table 5.2: Quantitative comparison of 3DLST on the airborne MS-LiDAR dataset.

Methods	Average F_1 score (%)	mIoU (%)	OA (%)	Latency (ms)
PointNet++ [89]	72.1	58.6	90.1	322.6
DGCNN [127]	71.6	51.0	91.4	86.2
RSCNN [66]	73.9	56.1	91.0	158.7
GACNet [122]	67.7	51.0	90.0	277.8
AGConv [164]	76.9	71.2	93.3	312.5
SE-PointNet++ [43]	75.9	60.2	91.2	-
FR-GCNet [161]	78.6	65.8	93.6	-
PointTransformer2* [160]	80.5	73.6	93.1	285.7
PPT-Net* [40]	80.1	73.6	92.7	43.3
PatchFormer* [151]	82.4	77.8	93.1	62.9
Xiao et al. [137]	83.3	79.3	94.0	-
ResMRGCN-28* [52]	81.1	74.0	93.3	45.7
GCNAS [154]	88.1	82.3	95.2	-
PTv3* [134]	85.5	77.1	93.4	40.6
3DGTN [70]	88.6	82.1	95.2	217.4
DTA-Former	88.3	81.4	95.4	78.9
3DLST	**89.3**	**82.3**	**95.4**	**11.7**

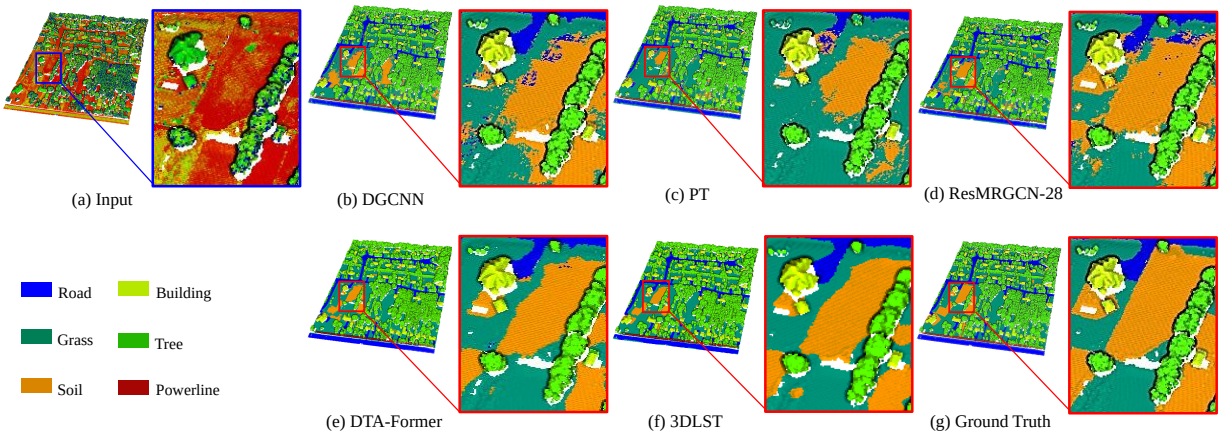


Figure 5.5: Comparison results from different methods on the testing area-11 in the airborne MS-LiDAR dataset.

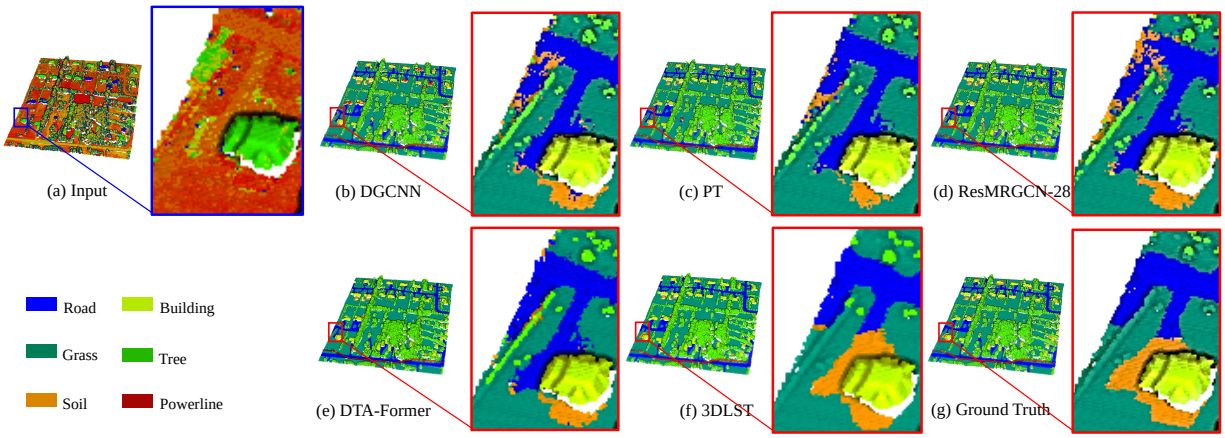


Figure 5.6: Comparison results from different methods on the testing area-12 in the airborne MS-LiDAR dataset.

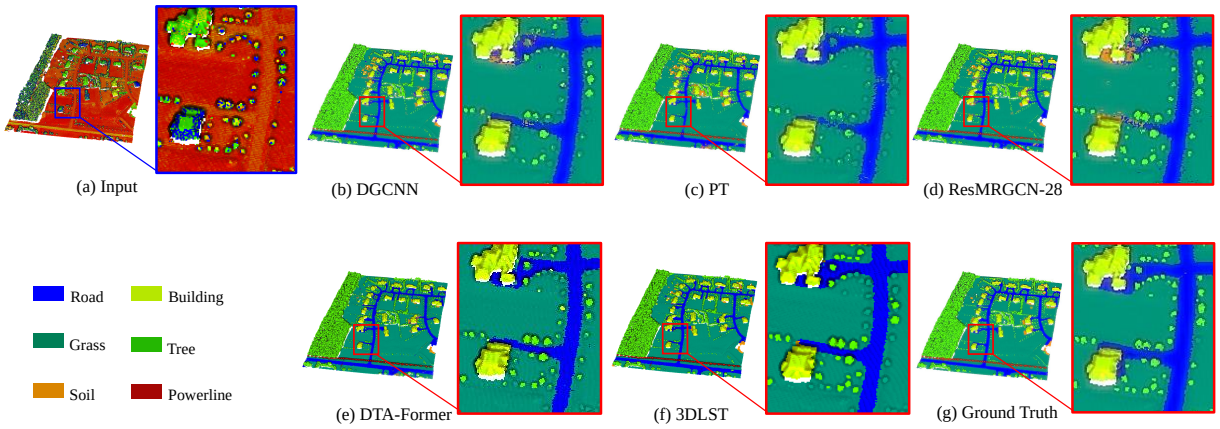


Figure 5.7: Comparison results from different methods on the testing area-13 in the airborne MS-LiDAR dataset.

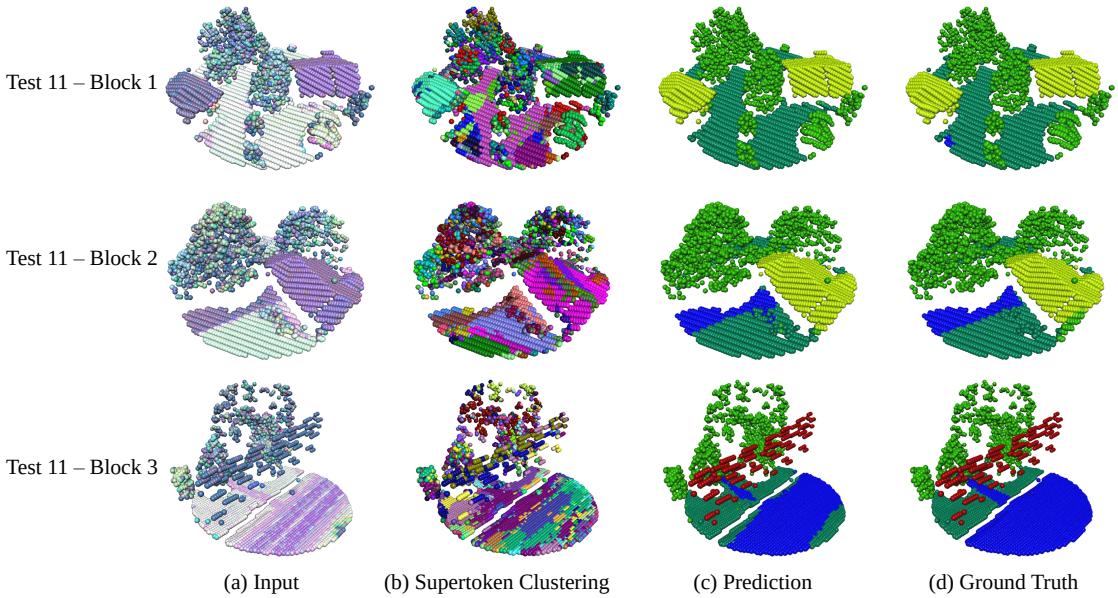


Figure 5.8: Visualization results of supertoken clustering and prediction on the testing area-11 blocks.

5.3.3 DALES Dataset

Dataset and Metrics. Dayton Annotated LiDAR Earth Scan (DALES) dataset is a comprehensive aerial LiDAR dataset designed for semantic segmentation, introduced by [119]. This dataset was collected using a Riegl Q1560 dual-channel system mounted on a Piper PA31 Panther Navajo aircraft. The collection covered 330 km^2 over Surrey, British Columbia, Canada, with DALES focusing on a 10 km^2 subset comprising 40 unique tiles, each spanning 0.5 km^2. The dataset features an average resolution of 50 points per square meter and maintains high accuracy through rigorous calibration and noise filtering. These characteristics make DALES particularly suitable for applications like 3D urban modelling, environmental monitoring, and deep learning algorithm benchmarking.

Each point in the DALES dataset includes 4 attributes: X, Y, Z coordinates, and intensity. The dataset encompasses eight semantic classes—Buildings, Cars, Trucks, Poles, Power lines, Fences, Ground, and Vegetation—meticulously hand-labelled with the aid of satellite imagery and a digital elevation model. DALES represents one of the largest publicly available aerial LiDAR datasets, providing over 500 million annotated points. It is split into training and testing subsets (70% and 30%, respectively) to facilitate robust

machine learning evaluations. For a fair comparison, each area is sampled using a 10 cm grid. The 40 areas are split into training and testing sets with roughly a 70/30 percentage. Furthermore, each area is divided into a series of $20m \times 20m$ blocks as training/testing samples, where each of them contained 8192 points after sampling. The mIoU and OA are used for performance evaluation.

Table 5.3: Quantitative segmentation comparison (%) on the DALES dataset.

Methods	OA	mIoU
PointNet++[89]	95.7	68.3
KPConv [117]	96.9	72.4
DGCNN[127]	96.1	67.6
ConvPoint[9]	97.2	67.4
SPG[49]	95.5	60.6
PointCNN[57]	97.2	58.4
ShellNet[158]	96.4	57.4
PT*[160]	97.0	73.4
SPT[99]	97.5	79.6
SuperCluster[100]	-	77.3
PReFormer[4]	92.9	70.9
PTv3*[134]	96.9	77.4
3DLST	**97.6**	**80.2**

Performance Comparison. Comparison results of different methods on the DALES dataset are shown in Table 5.3[7]. 3DLST is compared with other SOTA point cloud segmentation methods including the most recent superpoint Transformer method, SPT [99] and efficient Point Transformer V3 [134]. From the results, 3DLST achieves the best results compared to other benchmarked SOTA methods, with OA of 97.6% and mIoU of 80.2%. This demonstrates the excellent segmentation performance of 3DLST to aerial LiDAR data. Moreover, I also compared the inference speed of 3DLST and SPT [99] on one testing area (5135_54435) in detail. As shown in Table 5.4, with comparable accuracy, 3DLST is 1.6× faster than SPT in terms of the preprocessing step. This is because 3DLST does not require the time-consuming superpoint generation process. Since SPT utilized the traditional superpoint clustering as pre-processing, the size of input tokens to SPT is smaller than 3DLST. Therefore, the network inference speed of 3DLST is slightly lower than SPT. Overall, the entire inference speed of 3DLST is 1.6× faster than SPT, while

[7]Methods marked with * were not originally tested on this dataset in their respective papers. The reported results were obtained by running the official code provided by the authors.

achieving a slightly higher mIoU. This demonstrates that 3DLST achieves SOTA performance in both algorithm accuracy and speed. The visualization results of segmentation comparisons on the DALES dataset are shown in Figure 5.9, where the pink points represents the error points. They confirm the superiority of 3DLST in aerial LiDAR point cloud segmentation.

Table 5.4: Detailed comparison between SPT [99] and 3DLST on the DALES dataset.

Methods	OA (%)	mIoU (%)	Preprocessing (s)	Inference (s)
SPT[99]	97.5	79.6	125.3	**2.0**
3DLST	**97.6**	**80.2**	**76.1**	2.36

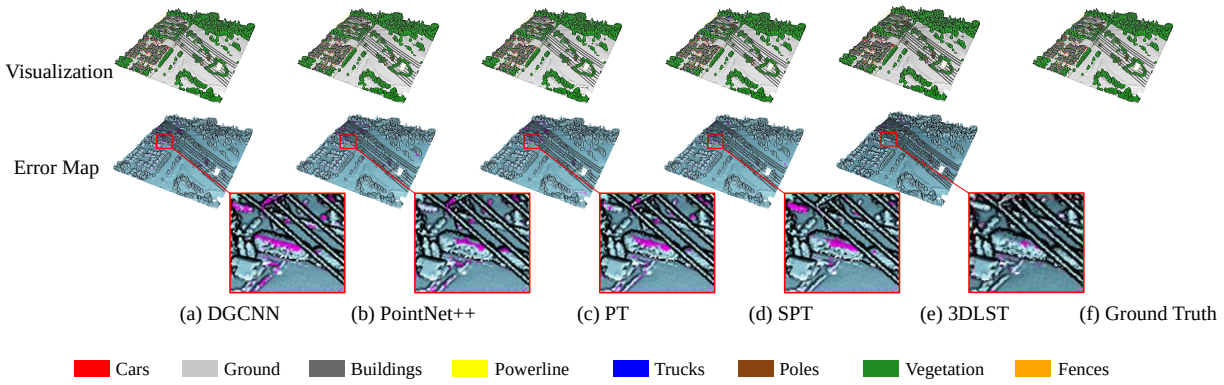


Figure 5.9: Comparison results from different methods on the DALES dataset.

5.3.4 Ablation Study

A series of ablation studies were conducted for the main blocks and parameters of 3DLST, to verify the effectiveness of the proposed blocks and robustness of the network to parameters. These experiments were performed on the airborne MS-LiDAR dataset.

Dynamic Supertoken Optimization. As described in Section 5.2.2, a cross-attention mechanism with a hard assignment is proposed for learnable supertoken optimization. Firstly, in the process of CAM generation, I replaced the $argmax$ operator in Eq. 5.2 with $softmax$. As shown in Table 5.5 Row 2, average F_1 score, mIoU, and OA values dropped 2.2, 2.4, and 0.8 absolute percentage points respectively after replacing. Compared with $softmax$, $argmax$ focuses on discrete decision-making, which is more suitable for the proposed supertoken clustering.

Table 5.5: Results (%) of ablation study on the airborne MS-Lidar dataset.

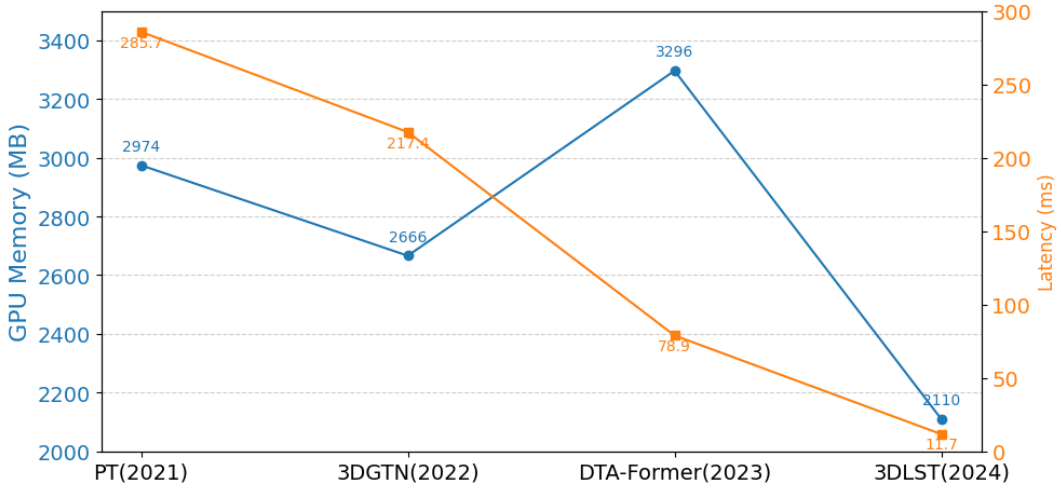
Ablation		Average F_1 (%)	mIoU (%)	OA (%)	Latency (ms)
DSO	$argmax \rightarrow softmax$	87.1	79.9	94.6	11.5
Architecture	W-net $\rightarrow U$-net	82.6	70.9	91.2	**11.3**
3DLST		**89.3**	**82.3**	**95.4**	11.7

Secondly, I tested the sensitivity of networks to the number S of learnable supertokens. As shown in Table 5.6, the best performance of 3DLST on the airborne MS-LiDAR dataset is achieved when S is set to 512. On the one hand, the smaller S may cause the network to ignore detailed features in a scene, resulting in inaccurate boundary segmentation and category misclassification. On the other hand, increasing S tends to limit the ability of the network to capture sufficient contextual information.

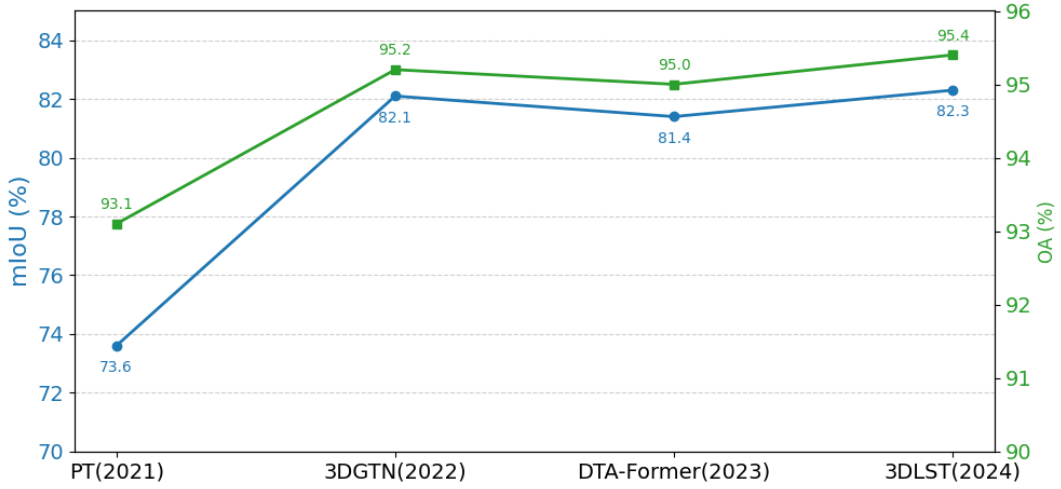
Table 5.6: Parameter sensitivity (%) of 3DLST to the number S of learnable supertokens on the airborne MS-LiDAR dataset.

S	Average F_1 score	mIoU	OA
128	80.1	69.6	93.1
256	87.8	80.3	94.5
512	**89.3**	**82.3**	**95.4**
1024	86.4	78.1	93.9

W-net Architecture. Table 5.5 Row 5 shows the performance comparison of W-net and U-net [89] architectures in 3DLST. From the results, the performance of the U-net drops significantly compared with the W-net, which is similar to the related ablation study in DTA-Former (Chapter 4). This is because the CAM-based token reconstruction in the U-net is lagging and inaccurate. In the U-net, the CAM used for feature pooling in the encoder is not directly reused or updated in the corresponding decoder stages, leading to lagging and less accurate token reconstruction. By contrast, the proposed W-net design permits immediate feedback from the CAM within the same module, ensuring the timeliness of feature interaction and reconstruction. In addition, we employ a multi-loss strategy in the W-net, computing loss for each module, which more effectively supervises the optimization of the corresponding model parameters.



(a) Comparison in Space and Time complexity



(b) Comparison in segmentation OA and mIoU

Figure 5.10: Memory Comparison of the proposed methods on the MS-LiDAR dataset.

5.4 Discussion

To further optimize the time complexity of DTA-Former while reducing its space complexity, the 3D Learnable Supertoken Transformer (3DLST) is proposed. Compared with DTA-Former, 3DLST adopts a Dynamic Supertoken Optimization (DSO) block that de-

fines learnable supertokens as clustering centers, which are dynamically optimized according to multi-level deep features. This design avoids both the information loss caused by discarding low-score points in the LTS block and the additional time overhead of sampling point selection. Unlike static superpoint strategies operating on the initial point features, the DSO block focuses on semantic homogeneity-aware token clustering, enabling supertokens to adaptively capture richer context information.

Additionally, 3DLST introduces an efficient Cross-Attention Upsampling (CAU) block that fully exploits long-range context dependency for accurate token reconstruction. Both DSO and CAU blocks are still deployed within the W-net architecture to better match Transformer-based feature learning. By addressing these limitations of DTA-Former, 3DLST delivers a more effective and efficient solution for large-scale LiDAR point cloud processing.

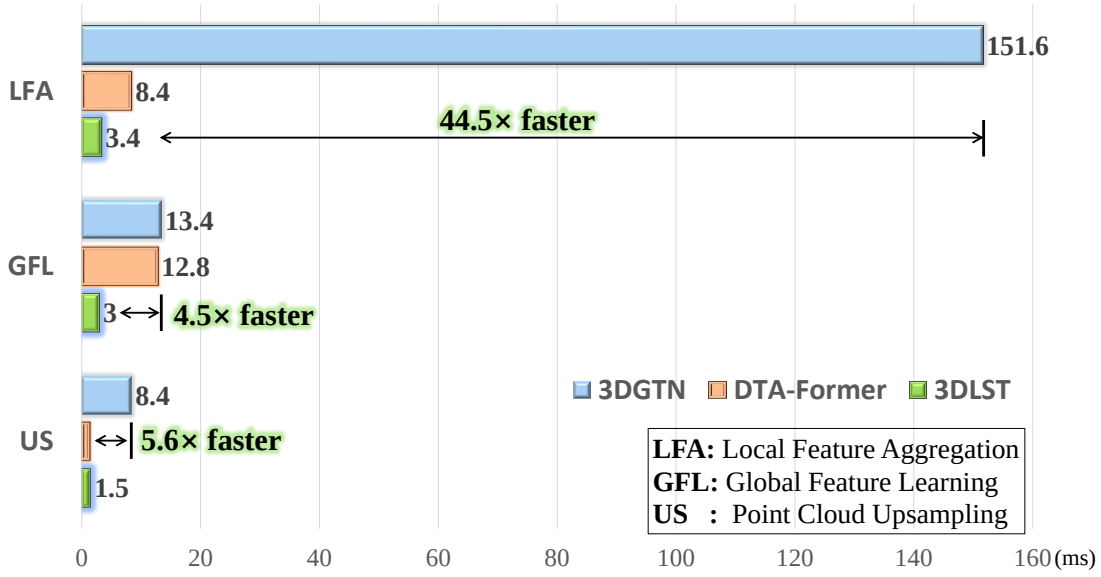


Figure 5.11: Latency comparison of the proposed methods on the MS-LiDAR dataset.

Semantic segmentation of a large-scale LiDAR point cloud covering an area of $500m \times 500m$ *with over 10 million points can be completed by 3DLST in under 1.5 minutes on a single GeForce RTX 2080 GPU (including data preprocessing, the semantic segmentation process, and data post-processing).* Figure 5.10 illustrates the iterative optimization and progressive improvements achieved by the three frameworks proposed in this thesis. In comparison with the pioneering 3D Transformer work PT [160], the efficiency and accuracy of the proposed models continue to increase. Figure 5.11 further confirms the progress made in developing efficient Transformer frameworks, and Figure 5.12 shows that 3DLST achieves

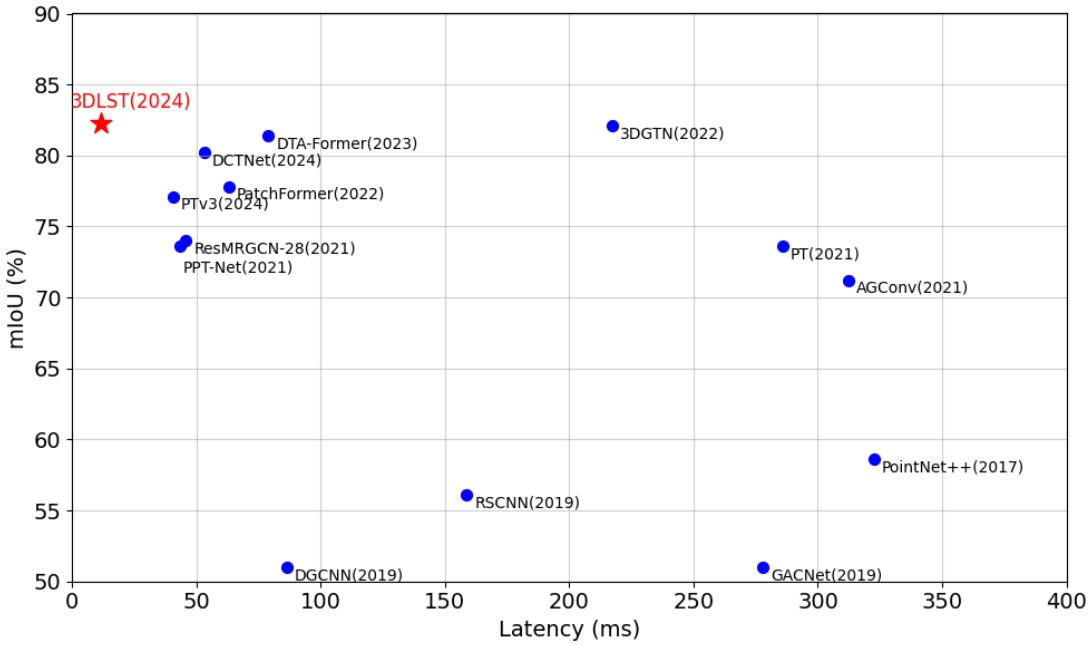


Figure 5.12: Comparison results of different methods on the MS-LiDAR dataset.

state-of-the-art (SOTA) performance on LiDAR point cloud data, surpassing other recent efficient Transformers such as PTv3 [134].

5.5 Chapter Summary

In this Chapter, an efficient Transformer framework, 3DLST, was presented to further enhance the performance of DTA-Former in both time and space complexity. Specifically, a novel Dynamic Supertoken Optimization (DSO) block was introduced to generate learnable supertokens, reducing information loss and avoiding the time-consuming sampling selection. Moreover, the Cross-Attention Upsampling (CAU) block was employed for precise token reconstruction by leveraging long-range context modeling. Both DSO and CAU blocks were integrated into the W-net architecture to accommodate hierarchical feature learning. Experimental comparisons on two challenging LiDAR datasets (airborne MS-LiDAR and DALES) demonstrated that 3DLST outperformed existing state-of-the-art methods in terms of efficiency and accuracy, while also exhibiting strong adaptability to multi-source LiDAR data.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This thesis aims to fully explore the potential of Transformer models in accurate and efficient LiDAR point cloud processing, by addressing the following key issues:

(1) How to design a Transformer-based point cloud representation framework, which accommodates key characteristics of LiDAR point clouds.

(2) Based on Issue 1, how to develop a fast Transformer framework that avoids the time-consuming operations in traditional hierarchical structures, while maintaining its excellent model accuracy.

(3) Bsed on Issue 2, how to further optimize the framework to overcome key drawbacks of Transformers (large computational complexity and memory consumption) for model efficiency improvement, especially in large-scale LiDAR point cloud scenarios.

The contributions are listed below correspondingly.

(1) A novel 3D Dual-attention Global-Local (GLocal) Transformer Network (3DGTN) is proposed, to address key challenges in LiDAR point cloud data. By capturing both global and local context, it enhances the representation capabilities of Transformers, ensuring its superior performance in point cloud processing.

(2) A fast point Transformer network with Dynamic Token Aggregation (DTA-Former) is developed to enhance model latency. It addresses inefficiencies in 3DGTN, such as point sampling, grouping, and reconstruction, while preserving the strong accuracy.

(3) A 3D Learnable Supertoken Transformer (3DLST) is introduced based on DTA-Former, further reducing the time and memory overhead of Transformer models. By adopting a novel supertoken clustering strategy, 3DLST not only lowers the number of input tokens but also enables semantic-homogeneity-aware token aggregation for enhanced feature representation. It achieves state-of-the-art results on multi-source LiDAR point clouds in terms of both accuracy and efficiency.

6.2 Recommendations for Future Research

In large-scale LiDAR point cloud segmentation and recognition, deep learning networks typically rely on extensive manual annotations. This reliance not only demands substantial human labor but also restricts the model's ability to generalize to unseen datasets. When dealing with vast and complex point clouds (e.g., urban or natural environments), obtaining labeled data becomes increasingly challenging, highlighting the potential value of unsupervised learning approaches.

Unsupervised methods aim to extract intrinsic patterns and relationships in point clouds without relying on ground truth labels. By leveraging geometric and structural cues, such methods can capture underlying semantics more flexibly, making them well-suited for diverse, high-dimensional datasets. Moreover, by removing the annotation requirement, unsupervised frameworks are more scalable and adaptable across various datasets.

This thesis demonstrates the strong capability of Transformer-based models for point cloud feature representation, indicating a promising direction toward unsupervised learning. For instance, GrowSP [159] uses an iterative process to refine a sparse convolutional network with pseudo-labels derived from geometric features. Incorporating Transformers into such unsupervised frameworks may further improve performance by harnessing their ability to capture long-range dependencies and intricate spatial relationships.

Considering the limited research on unsupervised deep learning for point cloud semantic segmentation, there remains significant scope for exploration. Introducing Transformers into unsupervised frameworks, combined with optimized training strategies, is an avenue well worth future investigation.

References

[1] Rashid Abbasi, Ali Kashif Bashir, Hasan J Alyamani, Farhan Amin, Jaehyeok Doh, and Jianwen Chen. Lidar point cloud compression, processing and learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 24(1):962–979, 2022.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[3] Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: encoding long and structured inputs in transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, 2020.

[4] Perpetual Hope Akwensi, Ruisheng Wang, and Bo Guo. PReFormer: A memory-efficient transformer for point cloud semantic segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 128:103730, 2024.

[5] Morris Antonello, Daniel Wolf, Johann Prankl, Stefano Ghidoni, Emanuele Menegatti, and Markus Vincze. Multi-view 3D entangled forest for semantic segmentation and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1855–1862. IEEE, 2018.

[6] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics*, 37(4):71, 2018.

[7] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detec-

tion with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1090–1099, 2022.

[8] Axel Berg, Magnus Oskarsson, and Mark O'Connor. Points to Patches: Enabling the use of self-attention for 3D shape recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 528–534. IEEE, 2022.

[9] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 88:24–34, 2020.

[10] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–14, 2014.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

[12] Chuanfa Chen, Huiming Wu, Ziming Yang, and Yanyan Li. Adaptive coarse-to-fine clustering and terrain feature-aware-based method for reducing liDAR terrain point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 200:89–105, 2023.

[13] Jiajing Chen, Burak Kakillioglu, and Senem Velipasalar. Background-aware 3-D point cloud segmentation with dynamic point feature aggregation. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.

[14] Jingdao Chen and Yong Kwon Cho. Crackembed: Point feature embedding for crack segmentation from disaster site point clouds with anomaly detection. *Advanced Engineering Informatics*, 52:101550, 2022.

[15] Xiaoxue Chen, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. PQ-transformer: Jointly parsing 3D objects and layouts from point clouds. *IEEE Robotics and Automation Letters*, 7(2):2519–2526, 2022.

[16] Hui-Xian Cheng, Xian-Feng Han, and Guo-Qiang Xiao. TransRVNet: LiDAR semantic segmentation with transformer. *IEEE Transactions on Intelligent Transportation Systems*, 24(6):5895–5907, 2023.

[17] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. PRA-Net: Point relation-aware network for 3D point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021.

[18] Yuwei Cheng, Jingran Su, Mengxin Jiang, and Yimin Liu. A novel radar point cloud generation method for robot environment perception. *IEEE Transactions on Robotics*, 38(6):3754–3773, 2022.

[19] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, Jun. 2019.

[20] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017.

[21] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

[22] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2978–2988, 2019.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–21, 2021.

[24] Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145, 2016.

[25] Mingtao Feng, Liang Zhang, Xuefei Lin, Syed Zulqarnain Gilani, and Ajmal Mian. Point attention network for semantic segmentation of 3D point clouds. *Pattern Recognition*, 107:107446, 2020.

[26] Yongbin Gao, Xuebing Liu, Jun Li, Zhijun Fang, Xiaoyan Jiang, and Kazi Mohammed Saidul Huq. LFT-Net: Local Feature Transformer Network for Point Clouds Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 24(2):2158–2168, 2023.

[27] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3809–3820. PMLR, 2021.

[28] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018.

[29] Bo Guo, Liwei Deng, Ruisheng Wang, Wenchao Guo, Alex Hay-Man Ng, and Wenfeng Bai. Mctnet: Multiscale cross-attention-based transformer network for semantic segmentation of large-scale point cloud. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2023.

[30] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

[31] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Voint Cloud: Multi-View Point Cloud Representation for 3D Understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–31, 2021.

[32] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:87–110, 2023.

[33] Wei Han, Xiaohan Zhang, Yi Wang, Lizhe Wang, Xiaohui Huang, Jun Li, Sheng Wang, Weitao Chen, Xianju Li, Ruyi Feng, et al. A survey of machine learning and deep learning in remote sensing of geological environment: Challenges, advances, and opportunities. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:87–113, 2023.

[34] Xian-Feng Han, Zhang-Yue He, Jia Chen, and Guo-Qiang Xiao. 3CROSSNet: Cross-level cross-scale cross-attention network for point cloud representation. *IEEE Robotics and Automation Letters*, 7(2):3718–3725, 2022.

[35] Xian-Feng Han, Yi-Fei Jin, Hui-Xian Cheng, and Guo-Qiang Xiao. Dual transformer for point cloud analysis. *IEEE Transactions on Multimedia*, 25:5638–5648, 2023.

[36] Xu Han, Zhen Dong, and Bisheng Yang. A point-based deep learning network for semantic segmentation of mls point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:199–214, 2021.

[37] Xu Han, Chong Liu, Yuzhou Zhou, Kai Tan, Zhen Dong, and Bisheng Yang. WHU-Urban3D: An urban scene lidar point cloud dataset for semantic instance segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 209:500–513, 2024.

[38] R Hassan, MM Fraz, A Rajput, and M Shahzad. Residual learning with annularly convolutional neural networks for classification and segmentation of 3D point clouds. *Neurocomputing*, 526:96–108, 2023.

[39] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11108–11117, 2020.

[40] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid Point Cloud Transformer for Large-Scale Place Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6098–6107, 2021.

[41] Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2017.

[42] Tengping Jiang, Yongjun Wang, Shan Liu, Qinyu Zhang, Lin Zhao, and Jian Sun. Instance recognition of street trees from urban point clouds using a three-stage neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:305–334, 2023.

[43] Zhuangwei Jing, Haiyan Guan, Peiran Zhao, Dilong Li, Yongtao Yu, Yufu Zang, Hanyun Wang, and Jonathan Li. Multispectral LiDAR point cloud classification using SE-PointNet++. *Remote Sensing*, 13(13):2516, 2021.

[44] Chaitanya Kaul, Joshua Mitton, Hang Dai, and Roderick Murray-Smith. Convolutional point transformer. In *Asian Conference on Computer Vision (ACCV)*, pages 303–319, 2022.

[45] Chaitanya Kaul, Nick Pears, and Suresh Manandhar. FatNet: A feature-attentive network for 3D point cloud processing. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 7211–7218, Jan. 2020.

[46] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3D semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–535. Springer, 2020.

[47] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified Transformer for 3D Point Cloud Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8500–8509, 2022.

[48] Loic Landrieu and Guillaume Obozinski. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4):1724–1766, 2017.

[49] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4558–4567, 2018.

[50] Guohao Li, Matthias Müller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. DeepGCNs: Making GCNs Go as Deep as CNNs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6923–6939, 2023.

[51] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9267–9276, 2019.

[52] Guohao Li, Matthias Müller, Guocheng Qian, Itzel C. Delgadillo, Abdulellah Abualshour, Ali Thabet, and Bernard Ghanem. DeepGCNs: Making GCNs go as deep as CNNs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6923–6939, 2023.

[53] Jincheng Li, Zhenxin Zhang, Haili Sun, Si Xie, Jianjun Zou, Changqi Ji, Yue Lu, Xiaoxu Ren, and Liuzhao Wang. Gl-Net: Semantic segmentation for point clouds of shield tunnel via global feature learning and local feature discriminative aggregation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 199:335–349, 2023.

[54] Li Li, Nan Song, Fei Sun, Xinyi Liu, Ruisheng Wang, Jian Yao, and Shaosheng Cao. Point2Roof: End-to-end 3D building roof modeling from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193:17–28, 2022.

[55] Qipeng Li, Yuan Zhuang, Jianzhu Huai, Yiwen Chen, and Alper Yilmaz. An efficient point cloud place recognition approach based on transformer in dynamic environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 207:14–26, 2024.

[56] Wei Li, Xing Wang, Xin Xia, Jie Wu, Xuefeng Xiao, Min Zheng, and Shiping Wen. Sepvit: Separable vision transformer. *arXiv preprint arXiv:2203.15380*, 2022.

[57] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution On X-Transformed Points. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 828–838, 2018.

[58] Yanzheng Li, Yinan Wang, and Yinhua Liu. Three-Dimensional Point Cloud Segmentation Based on Context Feature for Sheet Metal Part Boundary Recognition. *IEEE Transactions on Instrumentation and Measurement*, 72:1–10, 2023.

[59] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1489–1500, 2023.

[60] Ying Li, Lingfei Ma, Zilong Zhong, Dongpu Cao, and Jonathan Li. TGNet: Geometric graph CNN on 3-D point cloud segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3588–3600, 2019.

[61] Manxi Lin and Aasa Feragen. DiffConv: Analyzing Irregular Point Clouds with an Irregular View. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 380–397, 2022.

[62] Yi-Chun Lin and Ayman Habib. Semantic segmentation of bridge components and road infrastructure from mobile LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 6:100023, 2022.

[63] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network. In *AAAI Conference on Artificial Intelligence*, pages 8778–8785, 2019.

[64] Yahui Liu, Bin Tian, Yisheng Lv, Lingxi Li, and Fei-Yue Wang. Point cloud classification using content-based transformer via clustering in feature space. *IEEE/CAA Journal of Automatica Sinica*, 11(1):231–239, 2024.

[65] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5239–5248, 2019.

[66] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8895–8904, 2019.

[67] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[68] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for Efficient 3D Deep Learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 963–973, 2019.

[69] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1200–1211, 2023.

[70] Dening Lu, Kyle Gao, Qian Xie, Linlin Xu, and Jonathan Li. 3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–13, 2024.

[71] Dening Lu, Qian Xie, Kyle Gao, Linlin Xu, and Jonathan Li. 3DCTN: 3D convolution-transformer network for point cloud classification. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24854–24865, 2022.

[72] Dening Lu, Qian Xie, Mingqiang Wei, Linlin Xu, and Jonathan Li. Transformers in 3D point clouds: A survey. *arXiv preprint arXiv:2205.07417*, 2022.

[73] Dening Lu, Jun Zhou, Kyle Yilin Gao, Jing Du, Linlin Xu, and Jonathan Li. Dynamic clustering transformer network for point cloud segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 128:103791, 2024.

[74] Ziwei Luo, Ziyin Zeng, Wei Tang, Jie Wan, Zhong Xie, and Yongyang Xu. Dense Dual-Branch Cross Attention Network for Semantic Segmentation of Large-Scale Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024.

[75] Lingfei Ma, Ying Li, Jonathan Li, Weikai Tan, Yongtao Yu, and Michael A Chapman. Multi-scale point-wise convolutional neural networks for 3D object segmentation from LiDAR point clouds in large-scale environments. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):821–836, 2019.

[76] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–15, 2022.

[77] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3D point cloud understanding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1578–1587, 2019.

[78] Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. ∞-former: Infinite memory transformer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5468–5485, 2022.

[79] Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Diffuser: Multi-view 2d-to-3d label diffusion for semantic scene segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13589–13595. IEEE, 2021.

[80] Daniel Maturana and Sebastian Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, Sep. 2015.

[81] Kirill Mazur and Victor Lempitsky. Cloud transformers: A universal approach to point cloud processing tasks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10695–10704, 2021.

[82] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2906–2917, 2021.

[83] AAM Muzahid, Wanggen Wan, Ferdous Sohel, Lianyao Wu, and Li Hou. Curvenet: Curvature-based multitask learning deep networks for 3D object recognition. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1177–1187, 2021.

[84] Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. Efficient Transformers with Dynamic Token Pooling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6403–6417, 2023.

[85] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 604–621. Springer, 2022.

[86] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16949–16958, 2022.

[87] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9277–9286, 2019.

[88] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[89] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, Dec. 2017.

[90] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1–13, 2022.

[91] Xiaoyong Qiang, Weibing He, Shengyi Chen, Qingzhe Lv, and Fang Huang. Hierarchical point cloud transformer: A unified vegetation semantic segmentation model for multisource point clouds based on deep learning. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–16, 2023.

[92] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2021.

[93] Shi Qiu, Saeed Anwar, and Nick Barnes. PU-Transformer: Point Cloud Upsampling Transformer. In *Asian Conference on Computer Vision (ACCV)*, pages 326–343, 2021.

[94] Shi Qiu, Saeed Anwar, and Nick Barnes. Dense-resolution network for point cloud classification and segmentation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3813–3822, Jan. 2021.

[95] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18942–18952, 2022.

[96] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 13937–13949, 2021.

[97] Gernot Riegler, Ali O. Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3577–3586, Jul. 2017.

[98] Beanbonyka Rim, Ahyoung Lee, and Min Hong. Semantic segmentation of large-scale outdoor point clouds by encoder–decoder shared mlps with multiple losses. *Remote Sensing*, 13(16):3121, 2021.

[99] Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3D Semantic Segmentation with Superpoint Transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 17149–17158, 2023.

[100] Damien Robert, Hugo Raguet, and Loic Landrieu. Scalable 3D Panoptic Segmentation With Superpoint Graph Clustering. *arXiv preprint arXiv:2401.06704*, 2024.

[101] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5575–5584, 2022.

[102] Philipp Schuegraf, Jie Shan, and Ksenia Bittner. PLANES4LOD2: Reconstruction of LoD-2 building models using a depth attention-based fully convolutional neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 211:425–437, 2024.

[103] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[104] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3D object detection with channel-wise transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2743–2752, 2021.

[105] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020.

[106] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020.

[107] Hamin Song, Kichun Jo, Jieun Cho, Youngrok Son, Chansoo Kim, and Kwangjin Han. A training dataset for semantic segmentation of urban point cloud map for intelligent vehicles. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187:159–170, 2022.

[108] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[109] Yupeng Song, Fazhi He, Linkun Fan, Jicheng Dai, and Qing Guo. Dsacnn: Dynamically local self-attention cnn for 3D point cloud analysis. *Advanced Engineering Informatics*, 54:101803, 2022.

[110] Uwe Stilla and Yusheng Xu. Change detection of urban objects using 3D point clouds: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:228–255, 2023.

[111] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018.

[112] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, Dec. 2015.

[113] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3D scene instance segmentation. In *AAAI Conference on Artificial Intelligence*, pages 2393–2401, 2023.

[114] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 202–203, 2020.

[115] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7537–7547, 2020.

[116] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):109:1–109:28, 2023.

[117] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6411–6420, 2019.

[118] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1588–1597, 2019.

[119] Nina Varney, Vijayan K Asari, and Quinn Graehling. DALES: A large-scale aerial LiDAR data set for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 186–187, 2020.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 6000–6010, 2017.

[121] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13520–13529, 2023.

[122] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10296–10305, 2019.

[123] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2569–2578, 2018.

[124] Xu Wang, Yi Jin, Yigang Cen, Tao Wang, Bowen Tang, and Yidong Li. LighTN: Light-weight transformer network for performance-overhead tradeoff in point cloud downsampling. *IEEE Transactions on Multimedia*, 27:832–847, 2025.

[125] Xue Wang and Peijun Li. Extraction of urban building damage using spectral, height and corner information from VHR satellite images and airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:322–336, 2020.

[126] Yameng Wang, Yi Wan, Yongjun Zhang, Bin Zhang, and Zhi Gao. Imbalance knowledge-driven multi-modal network for land-cover semantic segmentation using aerial images and LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202:385–404, 2023.

[127] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, Nov. 2019.

[128] Ruilong Wei, Chengming Ye, Yonggang Ge, Yao Li, and Jonathan Li. Dynamic graph attention networks for point cloud landslide segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 124:103542, 2023.

[129] Xin Wei, Ruixuan Yu, and Jian Sun. View-GCN: View-based graph convolutional network for 3D shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1850–1859, Jun. 2020.

[130] Congcong Wen, Xiang Li, Xiaojing Yao, Ling Peng, and Tianhe Chi. Airborne lidar point cloud classification with global-local graph attention convolution neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:181–194, 2021.

[131] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5333–5343, 2023.

[132] Lemeng Wu, Xingchao Liu, and Qiang Liu. Centroid transformers: Learning to abstract with attention. *arXiv preprint arXiv:2102.08606*, 2021.

[133] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, Jun. 2019.

[134] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4840–4851. IEEE, 2024.

[135] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[136] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12978–12995, 2023.

[137] Kai Xiao, Jia Qian, Teng Li, and Yuanxi Peng. Multispectral LiDAR Point Cloud Segmentation for Land Cover Leveraging Semantic Fusion in Deep Learning Network. *Remote Sensing*, 15(1):243, 2022.

[138] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. MLCVNet: Multi-Level Context VoteNet for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10447–10456, 2020.

[139] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Vote-based 3D object detection with context modeling and SOB-3DNMS. *International Journal of Computer Vision*, 129(6):1857–1874, 2021.

[140] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4606–4615, 2018.

[141] Guoquan Xu, Hezhi Cao, Yifan Zhang, Yanxin Ma, Jianwei Wan, and Ke Xu. Adaptive Channel Encoding Transformer for Point Cloud Analysis. In *International Conference on Artificial Neural Networks (ICANN)*, pages 1–13. Springer, 2022.

[142] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3173–3182, 2021.

[143] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018.

[144] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5589–5598, 2020.

[145] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3323–3332, Jun. 2019.

[146] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics*, 35(6):1–12, 2016.

[147] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 288–307. Springer, 2022.

[148] Tan Yu, Jingjing Meng, Ming Yang, and Junsong Yuan. 3D object representation learning: A set-to-set matching perspective. *IEEE Transactions on Image Processing*, 30:2168–2179, 2021.

[149] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19313–19322, 2022.

[150] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11101–11111, 2022.

[151] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11799–11808, 2022.

[152] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. PVT: Point-voxel transformer for point cloud learning. *International Journal of Intelligent Systems*, 37(12):11985–12008, 2022.

[153] Nan Zhang, Zhiyi Pan, Thomas H Li, Wei Gao, and Ge Li. Improving graph representation for point cloud segmentation via attentive filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1244–1254, 2023.

[154] Qi Zhang, Yuanxi Peng, Zhiwen Zhang, and Teng Li. Semantic Segmentation of Spectral Lidar Point Clouds Based on Neural Architecture Search. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023.

[155] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Starting from non-parametric networks for 3D point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5344–5353, 2023.

[156] Yanan Zhang, Jiaxin Chen, and Di Huang. CAT-Det: Contrastively augmented transformer for multi-modal 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 908–917, 2022.

[157] Zhiwen Zhang, Teng Li, Xuebin Tang, Xiangda Lei, and Yuanxi Peng. Introducing improved transformer to land cover classification using multispectral lidar point clouds. *Remote Sensing*, 14(15):3808, 2022.

[158] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1607–1616, 2019.

[159] Zihui Zhang, Bo Yang, Bing Wang, and Bo Li. GrowSP: Unsupervised Semantic Segmentation of 3D Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17619–17629, 2023.

[160] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021.

[161] Peiran Zhao, Haiyan Guan, Dilong Li, Yongtao Yu, Hanyun Wang, Kyle Gao, José Marcato Junior, and Jonathan Li. Airborne multispectral LiDAR point cloud classification with a feature Reasoning-based graph convolution network. *International Journal of Applied Earth Observation and Geoinformation*, 105:102634, 2021.

[162] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, Jun. 2016.

[163] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. PTTR: Relational 3D Point Cloud Object Tracking with Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8521–8530, 2022.

[164] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4965–4974, 2021.

[165] Wei Zhou, Qian Wang, Weiwei Jin, Xinzhe Shi, and Ying He. Graph transformer for 3d point clouds classification and semantic segmentation. *Computers & Graphics*, 124:104050, 2024.

[166] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018.

[167] Xianghong Zou, Jianping Li, Yuan Wang, Fuxun Liang, Weitong Wu, Haiping Wang, Bisheng Yang, and Zhen Dong. Patchaugnet: Patch feature augmentation-based heterogeneous point cloud place recognition in large-scale street scenes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 206:273–292, 2023.

APPENDICES

Appendix A

Datasets and Metrics

A.1 Results on the ScanObjectNN Dataset

Datasets. To further evaluate the performance of 3DGTN to the real-world data captured by LiDAR scanning, the ScanObjectNN [118] dataset was also used in experiments. There are $\sim 15,000$ 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. I used the hardest variant of the dataset (PB_T50_RS) and adopted the original training/testing split as in [118]. Similarly, each sample from ScanObjectNN was downsampled to 1024 points. Since point clouds in PB_T50_RS have no normal information, only the 3D coordinates of point clouds are taken as input.

Performance Comparison. For the ScanobjectNN dataset, 3DGTN also achieves competitive performance with the SOTA approaches. Especially, as shown in Table A.1 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.

Table A.1: Quantitative classification comparison (%) on the ScanObjectNN dataset.

Methods	Input Size	mAcc	OA
Other Learning-based Methods			
PointNet++[89]	1024	75.4	77.9
PointCNN[57]	1024	75.1	78.5
DGCNN[127]	1024	73.6	78.1
SimpleView[27]	1024	-	80.5
PRANet[17]	1024	79.1	82.1
PointMLP[76]	1024	**84.4**	85.7
RepSurf[95]	1024	83.1	**86.0**
Transformer-based Methods			
PCT[30]	1024	77.1	80.5
PT[160]	1024	78.2	80.8
GBNet[92]	1024	77.8	80.5
Point-MAE[85]	1024	-	85.2
Point-TnT[8]	1024	-	83.5
Point-BERT[149]	1024	-	83.1
3DGTN	1024	**83.2**	**85.8**

A.2 Metrics Calculation

This thesis utilizes five different point cloud datasets to comprehensively evaluate the performance of the proposed algorithms. To ensure fair comparisons, I adopted evaluation metrics that are commonly used in existing related methods for each dataset. The details are outlined as follows.

Overall Accuracy (OA). The metric of OA is used in all five datasets (the ModelNet40 [135] and ScanObjectNN datasets [118], ShapeNet [146], MS-LiDAR dataset [161], and DALES [119]) for performance evaluation, which is formulated as:

$$OA = \frac{T}{N}, \tag{A.1}$$

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, 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.

Mean Intersection over Union (mIoU). The metric of mIoU is used for segmen-

tation performance evaluation:

$$\begin{aligned} mIoU &= \frac{\sum_{i=1}^{Cls} mIoU_i}{Cls}, \\ mIoU_i &= \frac{\sum TP_i}{\sum TP_i + \sum FP_i + \sum FN_i}, \end{aligned} \tag{A.2}$$

where Cls is the number of categories in the dataset, and $\sum TP_i$ represents the sum of TP samples in class i. The definitions of TP, FP, TN, FN are shown in Table A.2, where True Positive (TP) represents the model correctly predicts an instance as positive when it is positive, False Positive (FP) represents that the model incorrectly predicts an instance as positive when it is negative. False Negative (FN) represents the model incorrectly predicts an instance as negative when it is positive. True Negative (TN) represents the model correctly predicts an instance as negative when it is negative.

Table A.2: Definition of TP, TN, FP, FN based on prediction and actual labels.

Prediction / Actual	Positive (True)	Negative (False)
Positive (Predicted)	True Positive (**TP**)	False Positive (**FP**)
Negative (Predicted)	False Negative (**FN**)	True Negative (**TN**)

Average F_1 Score. The metric of average F_1 score is used for segmentation performance evaluation:

$$AverageF_1 = \frac{\sum_{i=1}^{Cls} F_{1i}}{Cls}, \tag{A.3}$$

where F_{1i} is calculated as follows:

$$\begin{aligned} F_{1i} &= 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i}, \\ Precision_i &= \frac{\sum TP_i}{\sum TP_i + \sum FP_i}, \\ Recall_i &= \frac{\sum TP_i}{\sum TP_i + \sum FN_i}. \end{aligned} \tag{A.4}$$

Appendix B

Publications during the PhD Study

1. **Dening Lu**, Qian Xie, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. 3DCTN: 3D Convolution-Transformer Network for Point Cloud Classification. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24854-24865, 2022.

2. **Dening Lu**, Kyle (Yilin) Gao, Qian Xie, Linlin Xu and Jonathan Li. 3DGTN: 3-D Dual-Attention GLocal Transformer Network for Point Cloud Classification and Segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1-13, 2024.

3. **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Jing Du, Linlin Xu and Jonathan Li. Dynamic clustering transformer network for point cloud segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 128:103791, 2024.

4. **Dening Lu**, Shimon Schwartz, Linlin Xu, Mohammad Javad Shafiee, Norman G. Vinson, Krzysztof J. Czarnecki, and Alexander Wong. Integrating deep transformer and temporal convolutional networks for SMEs revenue and employment growth prediction. *Expert Systems with Applications*, 252:124129, 2024.

5. **Dening Lu**, Linlin Xu, Jun Zhou, Kyle (Yilin) Gao, Zheng Gong and Dedong Zhang. 3D-UMamba: 3D U-Net with state space model for semantic segmentation of multi-source LiDAR point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 136:104401, 2025.

6. **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. Efficient Point Transformer with Dynamic Token Aggregating for LiDAR Point Cloud Segmentation. *Expert Systems with Applications*, under review, submitted at Mar 17, 2025, arXiv preprint arXiv:2405.15827.

7. **Dening Lu**, Jun Zhou, Kyle (Yilin) Gao, Linlin Xu and Jonathan Li. 3DLST: 3D Learnable Supertoken Transformer for LiDAR Point Cloud Scene Segmentation. *International Journal of Applied Earth Observation and Geoinformation*, under revision, submitted at January 06, 2025, arXiv preprint arXiv:2405.15826.

8. **Dening Lu**, Kyle (Yilin) Gao, Jonathan Li, Dedong Zhang and Linlin Xu. Exploring Token Serialization for Mamba-based LiDAR Point Cloud Segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, under revision, submitted at January 21, 2025.

9. Kyle (Yilin) Gao, **Dening Lu**, Hongjie He, Linlin Xu Jonathan Li and Zheng Gong. Enhanced 3D Urban Scene Reconstruction and Point Cloud Densification using Gaussian Splatting and Google Earth Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 2025, doi: 10.1109/TGRS.2025.3536169.

10. Wang, Lanying, **Dening Lu**, Weikai Tan, Yiping Chen, and Jonathan Li. Tree Species Classififcation Using Deep Learning Based 3d Point Cloud Transformer on Airborne Lidar Data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 974-977, 2023, doi: 10.1109/IGARSS52108.2023.10282813.

11. Wang, Lanying, **Dening Lu**, Linlin Xu, Derek T. Robinson, Weikai Tan, Qian Xie, Haiyan Guan, Michael A. Chapman, and Jonathan Li. Individual tree species classification using low-density airborne multispectral LiDAR data via attribute-aware cross-branch transformer. *Remote Sensing of Environment*, 315:114456, 2024.

Appendix C

Copyright Acknowledgment for IEEE Materials

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of *University of Waterloo*'s products or services. Internal or personal use of this material is permitted.

If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please visit IEEE Permissions to learn how to obtain a License from RightsLink.

The use of IEEE copyrighted material in this thesis is in accordance with IEEE's policies and guidelines. The author has ensured that the material is used responsibly, with proper acknowledgment and without any modifications that would misrepresent the original work.