

# Transformer based Semantic Segmentation in Large-scale Urban Point Clouds focusing on Rare Classes

A thesis accepted by the Faculty of Aerospace Engineering and Geodesy of the  
University of Stuttgart in partial fulfilment of the requirements for the degree of  
Doctor of Engineering Sciences (Dr.-Ing.)

by

**Xinlong Zhang, M.Sc.**

born in Henan, China

main referee: Prof. Dr.-Ing. Uwe Sörgel  
co-referee: Prof. Dr. habil. Hans-Gerd Maas  
date of defense: January 14, 2026

Institute for Photogrammetry and Geoinformatics  
University of Stuttgart  
2026

This thesis was published online on:

<http://www.dgk.badw.de/publikationen/reihe-c-dissertationen.html>  
and <http://elib.uni-stuttgart.de>

# Contents

|                                                                           |           |
|---------------------------------------------------------------------------|-----------|
| <b>Abstract</b>                                                           | <b>5</b>  |
| <b>Kurzfassung</b>                                                        | <b>7</b>  |
| <b>List of Abbreviations</b>                                              | <b>10</b> |
| <b>List of Figures</b>                                                    | <b>11</b> |
| <b>List of Tables</b>                                                     | <b>13</b> |
| <b>1 Introduction</b>                                                     | <b>15</b> |
| 1.1 Motivation . . . . .                                                  | 15        |
| 1.2 Objectives and contributions . . . . .                                | 17        |
| 1.3 Structure and organization . . . . .                                  | 18        |
| <b>2 Related Work</b>                                                     | <b>19</b> |
| 2.1 Point cloud segmentation with traditional methods . . . . .           | 19        |
| 2.2 Point cloud segmentation with deep learning techniques . . . . .      | 22        |
| 2.3 Attention mechanism in semantic segmentation . . . . .                | 28        |
| 2.4 Semantic segmentation for urban point clouds . . . . .                | 30        |
| 2.4.1 Complexity of large-scale urban point clouds . . . . .              | 31        |
| 2.4.2 Fine-grained and imbalanced classes in urban point clouds . . . . . | 32        |
| <b>3 Methodology</b>                                                      | <b>35</b> |
| 3.1 Attention network for large-scale point clouds . . . . .              | 35        |
| 3.1.1 Local self-attention operator . . . . .                             | 36        |
| 3.1.2 Redundant positional encoding . . . . .                             | 39        |
| 3.1.3 Multi-head attention . . . . .                                      | 40        |
| 3.1.4 Hierarchical network architecture . . . . .                         | 42        |
| 3.2 Local context embedding for fine-grained segmentation . . . . .       | 46        |
| 3.2.1 Local surface features . . . . .                                    | 46        |

|          |                                                                       |            |
|----------|-----------------------------------------------------------------------|------------|
| 3.2.2    | Surface feature extractor . . . . .                                   | 48         |
| 3.2.3    | Local context embedding . . . . .                                     | 49         |
| 3.3      | Contextual joint augmentation for rare classes . . . . .              | 52         |
| 3.3.1    | Sub-scene masking augmentation . . . . .                              | 54         |
| 3.3.2    | Dual augmentation of rare classes . . . . .                           | 55         |
| 3.3.3    | Dynamic joint augmentation strategy . . . . .                         | 57         |
| 3.4      | Target-aware attention for rare class segmentation . . . . .          | 58         |
| 3.4.1    | Pseudo labels for rare targets . . . . .                              | 58         |
| 3.4.2    | Target feature encoding module . . . . .                              | 60         |
| 3.4.3    | Candidate-aware mask module . . . . .                                 | 61         |
| 3.4.4    | Target topology parsing module . . . . .                              | 63         |
| 3.4.5    | Target confidence constraint module . . . . .                         | 64         |
| 3.4.6    | Target focus aggregation module . . . . .                             | 65         |
| 3.4.7    | Details of the network architecture . . . . .                         | 67         |
| <b>4</b> | <b>Data &amp; Evaluation Metrics</b>                                  | <b>71</b>  |
| 4.1      | Dataset description . . . . .                                         | 71         |
| 4.2      | Evaluation criteria . . . . .                                         | 76         |
| <b>5</b> | <b>Experiments &amp; Results</b>                                      | <b>79</b>  |
| 5.1      | Experiment design . . . . .                                           | 79         |
| 5.2      | Implementation details . . . . .                                      | 80         |
| 5.3      | Results and analysis . . . . .                                        | 81         |
| 5.3.1    | Attention network for semantic segmentation of urban scenes . . . . . | 82         |
| 5.3.2    | Local context embedding for fine-grained segmentation . . . . .       | 90         |
| 5.3.3    | Contextual joint strategy for rare class augmentation . . . . .       | 97         |
| 5.3.4    | Target-aware attention for rare class segmentation . . . . .          | 103        |
| <b>6</b> | <b>Conclusion and Outlook</b>                                         | <b>119</b> |
| 6.1      | Conclusion . . . . .                                                  | 119        |
| 6.2      | Outlook . . . . .                                                     | 121        |

# Abstract

Automated semantic interpretation of three-dimensional scenes has increasingly drawn attention in the fields of autonomous driving, building information modeling, and robotics, where reliable scene understanding plays a fundamental role in enabling intelligent perception and decision-making. Point clouds, obtained through laser scanning or stereo image matching technologies, have been widely acknowledged as the most reliable data source for representing spatial environments, owing to their ability to provide accurate and detailed 3D information. With the advances of lightweight laser scanning devices and flexible acquisition platforms, the availability of massive and high-resolution 3D point clouds has significantly increased. This unprecedented scale and richness of data, however, also poses a strong demand for effective methods that are capable of comprehensively interpreting large-scale and fine-grained point cloud scenes. In particular, the diversity of object categories, the imbalanced class distribution, and the intricate contextual relationships in large-scale urban areas bring great challenges to 3D semantic segmentation.

This research aims to address these challenges by developing novel methods and techniques that specifically focus on the 3D semantic segmentation of rare classes in large-scale urban point clouds. Hereby, the work provides contributions on four major aspects: (i) the hierarchical attentional framework tailored for large-scale point cloud segmentation, (ii) the embedding method of local surface features for fine-grained segmentation, (iii) the contextual joint augmentation strategy to strengthen the representation of rare classes, and (iv) the target-aware learning designed to enhance the segmentation of rare classes.

To achieve effective 3D semantic segmentation of urban scenes, we proposed a novel framework that integrates the attention mechanism focusing on crucial regions within large-scale point clouds. To further improve the perception of local geometric structures, the framework incorporates the embedding method of surface features into the segmentation pipeline, thereby facilitating the robust representation of fine-grained geometry inherent in complex environments. Addressing the issue of imbalanced class distribution, particularly concerning rare classes, we designed a context-guided data augmentation strategy that selectively augments underrepresented classes under the

constraint of their semantic and spatial contextual relationships. Moreover, focusing on unique geometric cues that characterize rare objects, we further developed a target-aware network that adaptively modulates attention to rare categories, thereby maintaining its high sensitivity to these underrepresented classes.

The proposed methods were evaluated through experiments on different large-scale benchmark datasets collected from multiple platforms, i.e., airborne, mobile and terrestrial platforms. In our submission to the official Hessigheim 3D benchmark ranking, our framework achieved a mean F1-score of 83.8% and an overall accuracy of 90.5%. In particular, the F1-scores of rare classes, namely vehicles and chimneys, significantly exceeded the average performance of other published methods, with improvements of 32.0% and 32.5%, respectively. Additionally, comprehensive experimental analyses on three other open benchmark datasets, including the Paris-Lille-3D dataset, the Semantic3D dataset and the WHU-Urban3D dataset, further demonstrate the robustness and effectiveness of our rare class segmentation methods.

# Kurzfassung

Die automatisierte semantische Interpretation dreidimensionaler Szenen hat in den Bereichen autonomes Fahren, Building Information Modeling und Robotik zunehmend an Bedeutung gewonnen, da ein zuverlässiges Szenenverständnis eine grundlegende Rolle für die Realisierung intelligenter Wahrnehmungs- und Entscheidungsprozesse spielt. Punktwolken, die durch Laserscanning oder Stereobild-Matching-Technologien erzeugt werden, gelten weithin als die zuverlässigste Datenquelle zur Darstellung räumlicher Umgebungen, da sie präzise und detaillierte 3D-Informationen bereitstellen können. Mit den Fortschritten bei leichten Laserscanning-Geräten und flexiblen Erfassungsplattformen hat die Verfügbarkeit massiver und hochauflösender 3D-Punktwolken erheblich zugenommen. Dieses beispiellose Ausmaß und die hohe Detailfülle der Daten erfordern jedoch wirksame Methoden, die in der Lage sind, großskalige und feinstrukturierte Punktwolkenszenen umfassend zu interpretieren. Insbesondere die Vielfalt der Objektkategorien, die unausgeglichene Klassenverteilung und die komplexen kontextuellen Zusammenhänge in großräumigen urbanen Szenarien stellen erhebliche Herausforderungen für die 3D-semantische Segmentierung dar.

Diese Forschung zielt darauf ab, die genannten Herausforderungen durch die Entwicklung neuartiger Methoden und Techniken zu bewältigen, die sich speziell auf die 3D-semantische Segmentierung seltener Klassen in großskaligen urbanen Punktwolken konzentrieren. Dabei leistet die Arbeit Beiträge in vier wesentlichen Aspekten: (i) dem hierarchischen Aufmerksamkeitsrahmen für großflächige Punktwolkensegmentierung, (ii) der Einbettungsmethode lokaler Oberflächenmerkmale zur fein-granularen Segmentierung, (iii) der kontextuellen gemeinsamen Augmentierungsstrategie zur Stärkung der Repräsentation seltener Klassen, und (iv) dem zielgerichteten Lernen für die Segmentierung seltener Klassen.

Um eine effektive 3D-semantische Segmentierung urbaner Szenen zu erreichen, haben wir ein neuartiges Framework vorgeschlagen, das einen Aufmerksamkeitsmechanismus integriert, der sich auf entscheidende Regionen innerhalb großskaliger Punktwolken konzentriert. Zur weiteren Verbesserung der Wahrnehmung lokaler geometrischer Strukturen integriert das Framework die Einbettungsmethode von Oberflächenmerkmalen in die Segmentierungspipeline, was eine ro-

buste Repräsentation der fein-granularen Geometrien in komplexen Umgebungen ermöglicht. Zur Bewältigung des Problems unausgeglichener Klassenverteilungen, insbesondere im Hinblick auf seltene Klassen, haben wir eine kontextgesteuerte Datenaugmentierungsstrategie entwickelt, die unterrepräsentierte Klassen gezielt unter Berücksichtigung ihrer semantischen und räumlichen Kontextbeziehungen verstärkt. Darüber hinaus wurde ein zielbewusstes Netzwerk entwickelt, das die Aufmerksamkeit adaptiv auf seltene Kategorien lenkt, indem es sich an deren einzigartige geometrische Merkmale anpasst und somit eine hohe Sensitivität gegenüber diesen unterrepräsentierten Klassen aufrechterhält.

Die vorgeschlagenen Methoden wurden durch Experimente auf verschiedenen großskaligen Benchmark-Datensätzen bewertet, die von unterschiedlichen Plattformen, d. h. luftgestützt, mobil und terrestrisch, gesammelt wurden. In unserer Einreichung für das offizielle Ranking des Hessigheim 3D-Benchmarks erzielten wir eine durchschnittliche F1-Score von 83,8% und eine Gesamtgenauigkeit von 90,5%. Insbesondere übertrafen die F1-Scores der seltenen Klassen, nämlich Fahrzeuge und Schornsteine, den Durchschnitt anderer veröffentlichter Methoden deutlich, mit Verbesserungen von 32,0% bzw. 32,5%. Darüber hinaus zeigen umfangreiche experimentelle Analysen auf drei weiteren offenen Benchmark-Datensätzen, dem Paris-Lille-3D Datensatz, dem Semantic3D Datensatz und dem WHU-Urban3D Datensatz, die Robustheit und Effektivität unserer Methoden zur Segmentierung seltener Klassen.



# List of Abbreviations

| Abbreviation | Description                                                 | Page |
|--------------|-------------------------------------------------------------|------|
| ALS          | Airborne Laser Scanning                                     | 15   |
| CAM          | Candidate-Aware Mask                                        | 61   |
| CNN          | Convolutional Neural Network                                | 23   |
| DBSCAN       | Density-Based Spatial Clustering of Applications with Noise | 20   |
| FPS          | Farthest Point Sampling                                     | 25   |
| GNN          | Graph Neural Network                                        | 27   |
| HT           | Hough Transform                                             | 20   |
| IoU          | Intersection over Union                                     | 65   |
| KNN          | K-Nearest Neighbor                                          | 36   |
| LiDAR        | Light Detection and Ranging                                 | 15   |
| MLP          | Multi-Layer Perceptron                                      | 25   |
| MLS          | Mobile Laser Scanning                                       | 15   |
| MRF          | Markov Random Field                                         | 22   |
| NLP          | Natural Language Processing                                 | 28   |
| NMS          | Non-Maximum Suppression                                     | 65   |
| OA           | Overall Accuracy                                            | 76   |
| PCA          | Principal Component Analysis                                | 21   |
| RANSAC       | Random Sample Consensus                                     | 20   |
| ReLU         | Rectified Linear Unit                                       | 37   |
| RF           | Random Forest                                               | 22   |
| SCE          | Surface Contextual Embedding                                | 49   |
| SSCN         | Submanifold Sparse Convolutional Network                    | 24   |
| SVM          | Support Vector Machine                                      | 22   |
| TCC          | Target Confidence Constraint                                | 64   |
| TFA          | Target Focus Aggregation                                    | 65   |
| TFE          | Target Feature Encoding                                     | 60   |
| TLS          | Terrestrial Laser Scanning                                  | 15   |
| TTP          | Target Topology Parsing                                     | 63   |
| UAV          | Unmanned Aerial Vehicle                                     | 16   |
| ViT          | Vision Transformer                                          | 29   |

# List of Figures

|      |                                                                                    |    |
|------|------------------------------------------------------------------------------------|----|
| 2.1  | Illustration of the pooling operation and the attention aggregation . . . . .      | 28 |
| 3.1  | Schematic diagram of the local attention aggregation module. . . . .               | 38 |
| 3.2  | Illustration of the multi-head attention. . . . .                                  | 41 |
| 3.3  | Detailed architecture of the hierarchical attention network. . . . .               | 44 |
| 3.4  | Illustration of surface descriptors at different scales. . . . .                   | 47 |
| 3.5  | Illustration of the local surface context embedding. . . . .                       | 50 |
| 3.6  | Detailed architecture of the local context embedding network. . . . .              | 51 |
| 3.7  | Workflow of contextual joint augmentation. . . . .                                 | 53 |
| 3.8  | Illustration of the sub-scene masking augmentation. . . . .                        | 54 |
| 3.9  | Illustration of the dual augmentation of rare classes. . . . .                     | 56 |
| 3.10 | Workflow of the pseudo label generation for rare targets. . . . .                  | 59 |
| 3.11 | Illustration of the proposed target-aware aggregator. . . . .                      | 60 |
| 3.12 | Workflow of the proposed target-aware aggregator. . . . .                          | 62 |
| 3.13 | Illustration of the target confidence constraint module. . . . .                   | 63 |
| 3.14 | Workflow of target aggregators for rare classes. . . . .                           | 66 |
| 3.15 | Detailed network architecture of the proposed target-aware attention. . . . .      | 68 |
| 4.1  | Illustration of the Hessigheim 3D ALS dataset. . . . .                             | 72 |
| 4.2  | Illustration of the Paris-Lille-3D MLS dataset. . . . .                            | 73 |
| 4.3  | Illustration of the selected scenes from the Semantic3D TLS dataset. . . . .       | 74 |
| 4.4  | Illustration of the selected scenes from WHU-Urban3D MLS dataset. . . . .          | 76 |
| 5.1  | Illustration of the training batch generation on Hessigheim 3D dataset. . . . .    | 80 |
| 5.2  | Analytical framework of the proposed method. . . . .                               | 81 |
| 5.3  | Classification results of the attention network on Hessigheim 3D dataset. . . . .  | 82 |
| 5.4  | Detailed results of the attention network on Hessigheim 3D dataset. . . . .        | 83 |
| 5.5  | Classification results of the attention network on Paris-Lille-3D dataset. . . . . | 86 |
| 5.6  | Sensitivity analysis of input noise for the attention network. . . . .             | 89 |
| 5.7  | Classification results using SCE module on Hessigheim 3D dataset. . . . .          | 91 |
| 5.8  | Detailed results using SCE module on Hessigheim 3D dataset. . . . .                | 92 |

---

|      |                                                                                               |     |
|------|-----------------------------------------------------------------------------------------------|-----|
| 5.9  | Classification results using SCE module on Paris-Lille-3D dataset. . . . .                    | 93  |
| 5.10 | Classification results using contextual joint augmentation on Hessigheim 3D dataset. . . . .  | 98  |
| 5.11 | Detailed results using contextual joint augmentation on Hessigheim 3D dataset. . . . .        | 99  |
| 5.12 | Classification results using contextual joint augmentation on Paris-Lille-3D dataset. . . . . | 100 |
| 5.13 | Classification results of TaaNet model on Hessigheim 3D dataset. . . . .                      | 104 |
| 5.14 | Detailed results of TaaNet model on Hessigheim 3D dataset. . . . .                            | 105 |
| 5.15 | Classification results of TaaNet model on Paris-Lille-3D dataset. . . . .                     | 107 |
| 5.16 | Classification results of our TaaNet model on the Semantic3D dataset. . . . .                 | 109 |
| 5.17 | Classification results of TaaNet model on MLS-S in WHU-Urban3D dataset. . . . .               | 111 |
| 5.18 | Classification results of TaaNet model on ALS-S in WHU-Urban3D dataset. . . . .               | 112 |

# List of Tables

|      |                                                                                                                        |     |
|------|------------------------------------------------------------------------------------------------------------------------|-----|
| 4.1  | Class frequencies in the training and validation sets for the Hessigheim 3D dataset.                                   | 72  |
| 4.2  | Class frequencies in the training and validation sets for the Paris-Lille-3D dataset.                                  | 73  |
| 4.3  | Class frequencies in the training and validation sets for the Semantic3D dataset.                                      | 75  |
| 5.1  | Confusion matrix of the attention network on Hessigheim 3D dataset.                                                    | 84  |
| 5.2  | Confusion matrix of PointNet++ on Hessigheim 3D dataset.                                                               | 85  |
| 5.3  | Confusion matrix of the attention network on Paris-Lille-3D dataset.                                                   | 86  |
| 5.4  | Confusion matrix of PointNet++ on Paris-Lille-3D dataset.                                                              | 87  |
| 5.5  | Comparison of segmentation results on Hessigheim 3D dataset using the attention network with different modality inputs | 88  |
| 5.6  | Comparison of number of parameters, computation time, and GPU memory usage of PointNet++ and our attention network.    | 90  |
| 5.7  | Confusion matrix using SCE module on Hessigheim 3D dataset.                                                            | 93  |
| 5.8  | Confusion matrix using SCE module on Paris-Lille-3D dataset.                                                           | 94  |
| 5.9  | Confusion matrix using early-stage embedding on Hessigheim 3D dataset.                                                 | 95  |
| 5.10 | Comparison of segmentation results on Hessigheim 3D dataset using the attention network with different local scales    | 96  |
| 5.11 | Comparison of number of parameters, computation time, and GPU memory usage with and without SCE module.                | 97  |
| 5.12 | Confusion matrix using the contextual joint augmentation on Hessigheim 3D dataset.                                     | 99  |
| 5.13 | Confusion matrix using the contextual joint augmentation on Paris-Lille-3D dataset.                                    | 101 |
| 5.14 | Comparison of segmentation results using Hessigheim 3D dataset with different augmentation strategies                  | 102 |
| 5.15 | Confusion matrix of TaaNet on Hessigheim 3D dataset.                                                                   | 106 |
| 5.16 | Comparison of segmentation results on Hessigheim 3D dataset using TaaNet and other baseline methods                    | 106 |
| 5.17 | Confusion matrix of TaaNet on Paris-Lille-3D dataset.                                                                  | 108 |
| 5.18 | Comparison of segmentation results on Paris-Lille-3D dataset using TaaNet and other baseline methods                   | 108 |
| 5.19 | Confusion matrix of TaaNet on Semantic3D dataset.                                                                      | 110 |

|      |                                                                                                                         |     |
|------|-------------------------------------------------------------------------------------------------------------------------|-----|
| 5.20 | Comparison of segmentation results on Semantic3D dataset using TaaNet and other baseline methods . . . . .              | 110 |
| 5.21 | Confusion matrix of TaaNet on MLS-S in WHU-Urban3D dataset. . . . .                                                     | 113 |
| 5.22 | Comparison of segmentation results on MLS-S in WHU-Urban3D dataset using TaaNet and other baseline methods. . . . .     | 113 |
| 5.23 | Confusion matrix of TaaNet on ALS-S in WHU-Urban3D dataset. . . . .                                                     | 114 |
| 5.24 | Comparison of segmentation results on ALS-S in WHU-Urban3D dataset using TaaNet and other baseline methods. . . . .     | 115 |
| 5.25 | Generalizability evaluation of TaaNet and other baseline methods on MLS-W in WHU-Urban3D dataset. . . . .               | 116 |
| 5.26 | Ablation analysis regarding the effect of the target-aware aggregator. . . . .                                          | 117 |
| 5.27 | Comparison of number of parameters, computation time, and GPU memory usage of TaaNet and other baseline models. . . . . | 117 |

# Chapter 1

## Introduction

### 1.1 Motivation

With the rapid advancement of Light Detection and Ranging (LiDAR) and photogrammetric techniques, the acquisition of accurate three-dimensional (3D) structural information of real-world environments has become increasingly accessible. Such information is typically represented as 3D point clouds, which play a crucial role in a wide range of applications, including autonomous driving, urban planning, and building information modeling. A fundamental task in these applications is the interpretation of point-wise semantics, such as vehicles, pedestrians, or buildings, commonly referred to as point cloud semantic segmentation. However, the inherent characteristics of 3D point cloud data present significant challenges for the semantic segmentation task.

Firstly, the irregularity and sparsity of point cloud data distinguish it from two-dimensional (2D) image data. While 2D images are characterized by fixed pixel grids and consistent density distributions, point clouds, typically acquired through LiDAR scanning or stereo image matching, exhibit highly irregular point distributions. Different data acquisition platforms and systems significantly influence variations of density and spatial distribution of points. Regions closer to the sensor, or those containing objects with complex geometric surfaces, tend to have higher point density, whereas areas farther from the sensor or those outside the optimal field of view are more sparsely represented. This phenomenon is particularly evident in terrestrial laser scanning (TLS) using fixed tripod platform. Furthermore, when mobile laser scanning (MLS) systems are mounted on movable transport platforms like vehicles, ground-level point clouds are often captured with high density, but the data for upper parts of buildings or pedestrians may be sparse. In contrast, airborne laser scanning (ALS) integrated into aerial platforms like airplanes or unmanned aerial

vehicles (UAVs), cover large areas but tend to focus on horizontal surfaces from a bird’s-eye view, such as rooftops and tree canopies. This disparity in point density and irregular distribution across platforms complicates the task of semantic segmentation. When dealing with unevenly distributed point clouds, models often struggle to capture critical features in sparse regions, especially when small objects or rare categories are involved, such as road signs or municipal poles, which are crucial for some real-world applications. Many existing segmentation methods fail to maintain consistent performance across all categories when faced with these irregular distributions, particularly underperforming on rare categories.

Secondly, the complexity of 3D scenes adds another layer of difficulty to point cloud segmentation. In the early stages of point cloud segmentation, traditional methods primarily relied on manually designed geometric features and demonstrated relatively stable performance, particularly in smaller and structurally simpler scenes. However, real-world environments consist of a diverse array of objects that exhibit considerable spatial variability and are often accompanied by challenges such as occlusion and overlap. Especially in large-scale scenarios, the intricate geometric structures and interactions between objects further complicate fine-grained segmentation challenges. In such cases, manually crafted features often failed to accurately capture the detailed structures and contextual information necessary for effective segmentation. In recent years, the rapid development of deep learning technologies has led to significant advancements in point cloud segmentation. Compared to traditional methods, deep learning models can automatically learn multi-level feature representations from large datasets, thus overcoming the limitations of manually designed features. However, most of these cutting-edge approaches developed in the computer vision and robotic community have primarily focused on ground-level scans with limited spatial coverage or indoor scenes. The complexity inherent in large-scale point clouds, as well as the diversity of objects present in urban environments, has often been overlooked.

Thirdly, the issue of class imbalance is particularly pronounced in large-scale 3D scenes. Small objects, due to their limited physical surface areas and low natural occurrence in real-world scenarios, typically account for only a tiny fraction of the total points in large-scale point clouds. Despite their low representation, these small objects play a crucial role in certain applications, such as extracting substructures of building details like chimneys, or subdividing the broader class ”urban furniture” into sub-categories of interest (e.g., traffic lights, road signs) for certain applications. This detailed segmentation of semantic categories introduces two major issues. Firstly, since these objects are mostly small, their recognition relies on significantly fewer points compared to larger compound objects. Secondly, classes of particular interest might be rare in some cases, leading to the ignorance of minority classes in the framework of standard classification schemes, as algorithms tend to focus on the more frequent classes by minimizing overall loss. This underrepresentation issue is exacerbated in large-scale point clouds with highly imbalanced categories. In this research,

we define this issue as rare class segmentation, which refers to the challenge of effectively identifying and segmenting rare categories in the face of data imbalance and class sparsity. Common strategies to address the issue of class imbalance, such as weighted loss functions and tailored sampling or augmentation methods, are mostly employed in the field of point cloud segmentation. However, most of them have focused on constraining training data or models based on point-level distribution of classes, without considering the natural geometric topology of 3D rare classes. To the best of our knowledge, existing research primarily emphasizes on the semantic segmentation of majority categories, while specialized methods for rare classe segmentation in large-scale point clouds have not been thoroughly investigated.

## 1.2 Objectives and contributions

In this thesis, we present a semantic segmentation framework for large-scale urban point clouds, with a particular focus on addressing rare class segmentation and class imbalance issues. Specifically, the complexity and diversity of urban point cloud datasets make segmentation tasks highly challenging, especially in the segmentation of rare classes and fine-grained objects. To tackle these challenges, this study introduces a series of innovative methods that not only enhance the segmentation performance of majority categories but also strengthen the recognition of rare classes in complex urban environments. Additionally, the strengths and limitations of the proposed methods will be discussed in detail, as well as potential solutions for future research. The primary objectives of this thesis are as follows:

- (1) A multi-head attention layer is developed to effectively capture local spatial discrepancies in point clouds, complemented by a redundant positional embedding strategy to encode relative topological relationships. Additionally, multi-scale attentions are aggregated within the hierarchical architecture, enhancing scene-level structural representation while retaining point details across spatial resolutions.
- (2) A surface feature with strong semantic interpretation is deeply embedded into the attentional network architecture to address the limitations of weak local representations. This integration improves the model’s ability to capture intricate details, particularly for fine-grained structures in large-scale scenes, where local features frequently suffer from underrepresentation.
- (3) A contextual joint augmentation strategy for rare classes is introduced to alleviate class imbalance in large-scale point clouds, aimed at increasing the diversity of rare classes under constrained scenarios while improving the model’s generalization capability for these classes.
- (4) A novel target aggregator incorporating natural geometric topology is designed to parse the

structural characteristics of rare categories and embed the corresponding confidence into point-wise attention, enabling the self-adaptive modulation of attention toward rare classes.

- (5) Extensive experimental evaluations are conducted on various benchmarks collected from different platforms to validate the robustness and effectiveness of the proposed methods in addressing rare class segmentation and large-scale urban scenes.

### 1.3 Structure and organization

This thesis is organized as follows: Chapter 2 presents a comprehensive review of the related work in our research fields, which includes traditional point cloud segmentation methods, deep learning based semantic segmentation techniques, and the application of the attention mechanism in point cloud processing. In particular, we highlight the large-scale segmentation and rare class segmentation. Chapter 3 introduces the proposed segmentation methods for large-scale urban point clouds and rare classes, namely the hierarchical network based on the attention mechanism for large-scale point clouds, local embedding of surface features for fine-grained segmentation, context-joint augmentation of rare classes, and target-aware learning based on geometric structures for rare class segmentation. Chapter 4 describes the benchmark datasets used to evaluate the proposed methods and evaluation metrics employed to assess the performance. Chapter 5 provides a comprehensive overview of the experimental design, implementation details, results, and their subsequent analysis. Chapter 6 concludes the presented work and discusses the limitations and potential future research directions.

## Chapter 2

# Related Work

As a key step in understanding 3D scenes, semantic segmentation of point clouds aims to classify each point into a specific semantic category, providing a solid foundation for higher-level environmental perception tasks. Therefore, the automated process of point cloud segmentation has become increasingly essential. Early research primarily focused on traditional handcrafted feature-based methods, which rely on manually designed geometric and attribute features to distinguish between different semantic categories. However, with the breakthrough advancements of deep learning in 2D image processing, researchers have begun exploring its application in semantic segmentation of point clouds, enabling end-to-end learning of semantic information directly from 3D point cloud data. With the exponential growth of large-scale point cloud data, the development of accurate and efficient semantic segmentation methods for urban-scale scenes has become a critical research area. Urban-scale point clouds are characterized by their massive data volume, high complexity, and diverse semantic classes, which pose significant challenges to existing segmentation techniques. In this chapter, we begin with a review of traditional methods for 3D point cloud segmentation, analyzing their principles and limitations (Section 2.1). It then provides a comprehensive overview of recent deep learning-based segmentation techniques (Section 2.2), with a particular focus on state-of-the-art Transformer models based on the attention mechanism (Section 2.3). Finally, building on existing approaches, we explore the key challenges in semantic segmentation of large-scale urban point clouds and discuss potential research directions and solutions (Section 2.4).

### 2.1 Point cloud segmentation with traditional methods

Before deep learning emerged as the mainstream approach for point cloud segmentation, researchers developed numerous traditional approaches to tackle this challenging problem (Maas and Vossel-

man, 1999; Lee and Schenk, 2002; Belton and Lichti, 2006; Brodu and Lague, 2012; Weinmann et al., 2013, 2015), establishing the foundation for modern point cloud processing techniques.

In the early stages, unsupervised methods were widely employed, leveraging geometric features, topological relationships, and statistical properties to group spatially adjacent points with similar characteristics (Morsdorf et al., 2004; Rabbani et al., 2006; Schnabel et al., 2007a; Golovinskiy and Funkhouser, 2009; Wang and Shan, 2009; Deschaud and Goulette, 2010). Clustering techniques partition point clouds based on feature similarity or spatial proximity, without necessarily imposing geometric model constraints (Nguyen and Le, 2013; Feng et al., 2014). These methods offer flexibility in handling complex, irregular structures that may not conform to predefined geometric primitives. K-means clustering represents one of the most straightforward partitioning approaches, iteratively assigning points to the nearest cluster center and updating center positions until convergence. Morsdorf et al. (2004) applied K-means clustering to airborne LiDAR data using features derived from height distributions and echo characteristics, segmenting individual tree crowns in forested areas. While computationally efficient, it requires pre-specifying the number of clusters and tends to produce spherical clusters, limiting its efficacy for irregularly shaped segments. Ester et al. (1996) introduced a clustering algorithm, namely density-based spatial clustering of applications with noise (DBSCAN), which groups points based on their local density effectively identifying clusters of varying shapes and sizes. Rusu et al. (2008) extended it for 3D segmentation through the incorporation of surface normal consistency constraints, effectively preserving geometric boundaries between adjacent objects. Their approach demonstrated remarkable robustness in cluttered indoor environments with complex object arrangements. Although these methods are computationally efficient, they often struggle with robustness in scenarios where point cloud density varies significantly. Another branch consists of rule-based methods, which attempt to fit geometric primitives or mathematical models to point cloud data to identify specific structures within the scene (Rabbani and Van Den Heuvel, 2005; Schnabel et al., 2007b; Tarsha-Kurdi et al., 2007; Pu and Vosselman, 2009). These approaches prove particularly effective when dealing with environments dominated by regular geometric shapes. Notable examples include the hough transform (HT) algorithm (Ballard, 1981) and the random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981), which have become cornerstone techniques for robust primitive segmentation in point clouds. Vosselman et al. (2004) introduced a modified HT for plane segmentation, utilizing the normal vector and plane-to-origin distance to reduce computational complexity and enhance robustness against noise and outliers. Schnabel et al. (2007a) developed an efficient RANSAC-based framework that sequentially extracts basic geometric primitives such as planes, spheres, cylinders, cones, and tori from noisy point cloud data. Their approach iteratively samples minimal point sets, constructs candidate shape hypotheses, and evaluates the fitness of these hypotheses against the remaining points. However, these methods are highly sensitive to parameter selection and strongly influenced by point cloud

characteristics such as density and noise (Vo et al., 2015). Region growing algorithms (Besl and Jain, 1988) constitute another prominent category of rule-based approaches, employing an aggregative strategy that progressively expands regions based on predefined similarity criteria (Rabbani et al., 2006; Zhou et al., 2014; Dong et al., 2018; Lin et al., 2019). Tóvári and Pfeifer (2005) used the normal vector, the distance between neighboring points and the fitted plane, and the distance between the current and candidate points as criteria for merging points into a randomly selected seed region. Rabbani et al. (2006) introduced a constrained region growing algorithm that initiates from seed points exhibiting minimum local surface variation. Then it expands regions by examining both normal vector deviation and spatial proximity between points, effectively segmenting smooth surfaces while preserving sharp geometric boundaries. Deschaud and Goulette (2010) proposed a voxel-based region growing method, replacing points with voxels and using an octree to efficiently merge regions during segmentation. Vo et al. (2015) enhanced traditional region growing techniques by incorporating an adaptive mechanism that dynamically adjusts similarity thresholds based on local surface characteristics. Their approach accommodates varying point densities and noise levels, resulting in improved segmentation consistency across heterogeneous scenes. While effective for specific applications such as building extraction and ground detection, these methods heavily depend on handcrafted features and exhibit limited generalizability, making them less adaptable to complex and dynamic environments.

To achieve the semantic labeling of point clouds, supervised segmentation methods leveraging handcrafted features combined with classical machine learning algorithms were typically employed as the primary strategy (Chehata et al., 2009; Shapovalov et al., 2010; Niemeyer et al., 2012; Hackel et al., 2016). Some studies focused on extracting discriminative feature representations that encapsulate the intrinsic geometric characteristics of 3D points, where the local context of each point is represented by various handcrafted mathematical expressions derived from its spatial neighborhood (Rusu et al., 2009; Tombari et al., 2010; Blomley et al., 2014). Based on principal component analysis (PCA), these approaches extract eigenvalues and derived features invariant to shift, scale, and rotation from the covariance matrix of local coordinates (i.e., structure tensor) (Gumhold et al., 2001). Building upon this foundation, Pauly et al. (2003) estimated the change of curvature for point clouds through eigenvalues, with their sum indicating local neighborhood variation. Lalonde et al. (2005) developed the specialized descriptors to capture the distinctive characteristics of different environmental elements, including vegetation, ground surfaces, and vertical structures. Demantké et al. (2011) proposed a dimensionality-based generic feature set that classifies local geometries as linear, planar, or volumetric structures based on the relative magnitudes of eigenvalues. Weinmann et al. (2013) introduced a seminal framework for extracting discriminative features from point clouds, proposing an eigenentropy-based approach for optimal neighborhood size selection. This technique adaptively determines the most appropriate scale for feature computation by analyzing

the entropy of eigenvalues derived from the local neighborhood’s covariance matrix. Following this work, Weinmann et al. (2015) further developed a comprehensive feature extraction pipeline, which computes a diverse set of descriptors capturing various geometric properties from the 3D structure tensor of each point’s local neighborhood. Others concentrated on classifier design to maximum the discrimination of features with various semantics, mapping the extracted features to specific categories. A variety of classifiers have been introduced and evaluated for the classification task of point clouds in previous work. For example, Secord and Zakhor (2007) explored the application of support vector machines (SVM) for urban tree detection by integrating LiDAR data and aerial imagery for classification. Chehata et al. (2009) demonstrated the efficacy of random forests (RF) for airborne point cloud classification, employing an ensemble of decision trees to handle high-dimensional feature spaces while maintaining computational efficiency. Kim et al. (2010) further validated the effectiveness of the RF classifier in classifying 3D power lines in urban scenes. In addition, graph structures can be constructed to encode spatial dependencies among 3D points, where nodes represent individual points or local point groups (supervoxels), and edges encode spatial relationships or feature similarities between connecting nodes. And graph partitioning techniques are then utilized to segment the point cloud into distinct regions. Anguelov et al. (2005) proposed a discriminative learning framework based on markov random fields (MRF) to segment point clouds in both indoor and outdoor environments, which jointly incorporates local geometric features and the spatial dependencies between neighboring points. Niemeyer et al. (2014) employed conditional random fields (CRF) to model both spatial and semantic dependencies among points, facilitating the integration of local and global contextual information for accurate point cloud classification and building detection in complex urban environments. Although these methods effectively capture local dependencies within the point cloud, their high computational complexity limits the applicability to large-scale scenes. Building upon these works, Weinmann et al. (2015) conducted an extensive comparative analysis of various classifiers and concluded that RF generally offers the optimal balance between computational complexity and segmentation accuracy. Their work also highlighted that the careful design of geometric features plays a crucial role in the overall segmentation performance. Nevertheless, the effectiveness of these methods are highly dependent on the discriminative power of the designed features and the ability of the chosen classifier to differentiate between distinct object categories or surface types.

## **2.2 Point cloud segmentation with deep learning techniques**

In recent years, the rapid advancement of depth sensors and 3D scanning technologies has established 3D point clouds as a fundamental spatial data representation, playing an indispensable role in various fields such as geospatial mapping, autonomous driving, and robotics (Nagy and Benedek, 2018; Shi et al., 2019; Bassier and Vergauwen, 2020; Chen et al., 2021; Grandio et al., 2022; Qiu

et al., 2024). Driven by deep learning techniques, significant breakthroughs have been achieved in the fields of computer vision and pattern recognition, particularly in 3D scene understanding (Su et al., 2015; Maturana and Scherer, 2015; Qi et al., 2017a,b; Thomas et al., 2019; Milioto et al., 2019; Zhao et al., 2021). Compared to traditional methods for point cloud segmentation, deep learning-based approaches can automatically extract high-level abstract features from raw point cloud data, which eliminate the need for complex hand-crafted feature design while enabling more accurate interpretation and differentiation of semantic categories in complex 3D environments. However, due to the irregular and unstructured characteristics of point cloud data, extending deep neural networks to handle 3D points remains a significant challenge (Guo et al., 2020; Cui et al., 2021; Zhang et al., 2023a; Sarker et al., 2024). Depending on the data representation and processing strategy, deep learning-based semantic segmentation methods for point clouds can generally be categorized into four groups, namely projection-based methods, voxel-based methods, point-based methods, and graph-based methods, which are discussed in the following.

## Projection-based methods

In order to leverage the well-established techniques and models developed for 2D images (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016), early deep learning-based methods projected irregularly distributed 3D point clouds onto 2D planes, converting them into a set of 2D rasterized representations (Feng et al., 2018; Tatarchenko et al., 2018; Aksoy et al., 2020; Alnaggar et al., 2021; Zhou et al., 2021; Huang et al., 2023). Feature learning and semantic segmentation were then performed on the 2D projections, and the predicted labels were subsequently back-projected to the original 3D space to achieve point-wise semantic labeling. Multi-view methods represent a prevalent class of projection-based approaches. These methods render 3D point clouds from multiple viewpoints to generate a series of 2D images, which are then processed using standard 2D convolutional neural networks (CNN). One notable work in this area is MVCNN (Su et al., 2015), which treats the combination of multi-view images as a compact representation of 3D geometric shape. Each rendered image is fed into a 2D CNN to extract features for object classification. Building on this, Lawin et al. (2017) extended the projection concept by transforming point clouds into multi-modal synthetic 2D representations, including color, depth, and surface normal images. These were then fed into a multi-stream 2D CNN for semantic segmentation to predict pixel-wise scores. Subsequently, the semantic label of each point was determined through a fusion strategy followed by a back-projection process. In Boulch et al. (2018), suitable snapshot images combining meshes and point clouds were generated from a RGB view and a depth composite view. The semantic labels for the 2D snapshots were predicted using 2D fully convolutional networks, and an efficient buffer-based back-projection mechanism was used to transfer the results to the 3D space, enabling effective segmentation of point clouds. Zhao et al. (2018) in-

roduced a multi-scale projection strategy that incorporated rich contextual information, such as height, intensity, and roughness, into the 2D images. A multi-scale CNN was then employed to classify ALS point clouds based on these derived representations. To alleviate the information loss from 3D-to-2D projection, You et al. (2019) proposed PVRNet to explicitly model the relationship between 3D points and their corresponding views, which fuses features via a correlation prediction module through single-view and multi-view fusion strategies. Range-view methods represent another branch of projection strategies, which typically project 3D point clouds onto cylindrical or spherical coordinate systems to generate range images. The projection process preserves certain 3D neighborhood structures while allowing for efficient processing using 2D CNNs. Wu et al. (2018) introduced the SqueezeSeg network based on spherical range images to achieve an end-to-end 3D segmentation pipeline. Inspired by this, Milioto et al. (2019) developed RangeNet++, which adopts spherical projections as an intermediate representation and incorporates GPU-accelerated post-processing for real-time point-wise classification. Although projection-based methods benefit from well-established 2D CNN architectures without substantial modifications to the network design, the projection process inevitably results in the loss of 3D spatial information, especially in scenes with complex geometries or occlusions. In addition, the other limitations such as compression of depth information and interpolation errors in back-projection process significantly affect the accuracy of segmentation results.

## **Voxel-based methods**

To address the inherent irregular and unstructured nature of point cloud data, another deep learning strategy involves converting 3D point clouds into regular voxel grids (Zhang et al., 2020; Tang et al., 2020; Cheng et al., 2021b; Liu et al., 2023). This voxelization process discretizes the continuous 3D space into a series of small cubic units (i.e., voxels), each of which may contain several points and can be represented through aggregation rules such as feature averaging. By converting point clouds into a structured representation, standard 3D CNNs can be directly applied for feature extraction and semantic segmentation. After segmentation, the voxel-level predicted labels can be back-mapped to the original points to achieve semantic assignment at the point level. One of the earliest representative works to introduce 3D CNNs for point cloud processing was VoxNet (Maturana and Scherer, 2015), which voxelized point clouds and extracted features using 3D convolutions, followed by 3D max pooling and fully connected layers for classification. VoxNet demonstrated the feasibility of applying 3D CNNs to point cloud data. To address the sparsity of point cloud distributions in 3D space, several studies have focused on architectural improvements. Notably, the sparse 3D CNNs (Graham, 2015) only performed convolutions on non-empty voxels, significantly reducing computational cost and memory usage. The submanifold sparse convolutional networks (SSCNs) (Graham et al., 2018) was a representative approach for 3D semantic segmen-

tation tasks, offering high computational efficiency while maintaining high accuracy. The follow-up study (Schmohl and Sörgel, 2019) further validated the effectiveness and robustness of SSCNs for large-scale ALS point cloud segmentation. To further improve the efficiency of the voxelization process, alternative 3D data structures such as Octrees (Wang et al., 2017) and Kd-trees (Klokov and Lempitsky, 2017) have been introduced to optimize spatial partitioning and memory storage. In Riegler et al. (2017), unbalanced octrees were utilized to adaptively partition point clouds into non-uniform voxels, which concentrates memory and computational resources on relevant regions of high density of occupied voxels, without sacrificing resolution. Liu et al. (2019b) proposed a dual-branch architecture that processes both point-wise and voxel-wise representations. The voxel branch aggregated coarse neighborhood features, while the point branch used a multilayer perceptron (MLP) to extract fine-grained point features, allowing the model to leverage the strengths of both representations. However, similar to the limitations of projection-based approaches, voxel-based methods inevitably introduce quantization errors during the voxelization process, leading to the loss of geometric details in the original point cloud. This trade-off is largely governed by the resolution setting: high-resolution grids preserve more spatial information but at the cost of increased computational and memory overhead, whereas low-resolution grids may fail to capture fine geometry. Moreover, semantic ambiguity can arise near voxel boundaries, where points from different semantic categories may be quantized into the same voxel and assigned identical labels. This ambiguity can affect both the training and inference stages, posing challenges to accurate scene understanding.

## Point-based methods

In contrast to approaches that convert unstructured point clouds into regular representations such as images or voxels, point-based methods directly take raw point cloud data as input to deep neural networks and perform feature extraction and classification at the point level. This strategy aims to retain the inherent structure and dispersion pattern of the points, avoiding potential information loss introduced by intermediate representations (Zhang et al., 2019; Guo et al., 2020; Sarker et al., 2024). Consequently, a range of deep learning-based techniques for direct point cloud processing have emerged in recent years, gradually becoming the mainstream solution for 3D semantic segmentation tasks. PointNet (Qi et al., 2017a) is a pioneering work in point-based methods, which developed a shared MLP to learn point-wise features and aggregated global features using symmetric pooling functions such as max pooling. The segmentation network then combined global features with point-wise features to achieve point-wise semantic predictions. To address the issues of lacking local topology, PointNet++ (Qi et al., 2017b) introduced a hierarchical architecture based on set abstraction modules, where point clouds were sampled using farthest point sampling (FPS) and local regions are grouped via nearest neighbor search progressively. Within each local group,

PointNet was applied to aggregate local features to capture detailed geometric structures. This hierarchical process is similar to CNNs in image processing, progressively expanding the receptive field and enhancing contextual understanding. To further enhance spatial context modeling, Engelmann et al. (2017) expanded PointNet by introducing input-level context to incorporate multi-scale or neighboring regions, and output-level context to share feature across larger spatial neighborhoods via consolidation units. In Engelmann et al. (2018), feature space clustering was introduced to integrate neighborhood information in Euclidean space, which produced robust feature representations for local regions. To improve the local representation capacity of point features without significantly increasing the network’s parameters, Jiang et al. (2018) inserted a scale-adaptive PointSIFT module into the hierarchical structure of PointNet++, encoding directional information from multiple orientations. SO-Net (Li et al., 2018b) constructed explicit spatial nodes for input points using a self-organizing map, preserving the topological properties of the input space. Yousefhusien et al. (2018) extended PointNet by integrating a multi-scale framework specifically tailored for ALS point clouds. To process large-scale point clouds, Hu et al. (2020) proposed RandLA-Net, a highly efficient and lightweight architecture that utilized random point sampling instead of FPS and aggregated features using local spatial encoding modules combined with attentional pooling units. Moreover, PointNeXt (Qian et al., 2022) is regarded as the next-generation extension of PointNet++, incorporating inverted residual bottleneck designs and more effective data augmentation and optimization techniques to significantly enhance performance. In addition, Ma et al. (2022) proposed PointMLP, a novel simplified framework that eliminates the complex set abstraction layers and feature transformation modules of PointNet++. Instead, it employs residual MLPs combined with geometry-aware local spatial encoding to capture detailed local features. To better exploit the local geometric relationships between points, another branch focused on defining convolution operations directly on point clouds (Xu et al., 2018; Liu et al., 2019a; Boulch, 2020; Xu et al., 2021b). Unlike 2D image convolutions performed on regular grids, point convolutions focus on defining local neighborhoods and designing convolution kernels adapted to local geometric structures. For example, Li et al. (2018c) proposed a new architecture for feature learning from point clouds, introducing the  $\chi$ -conv operator to weight and permute input points before standard convolutions. In Thomas et al. (2019), a set of kernel points was designed to define convolution kernels that adapt to diverse local geometries, where the contributions of kernel positions are determined through linear weighting. Wu et al. (2019) utilized MLPs to learn weight functions and applied density-adaptive reweighting to convolution kernels, mitigating the impact of non-uniform point distributions. To expand the receptive field, Engelmann et al. (2020) proposed dilated point convolutions, which aggregate features from dilated neighborhoods rather than considering only the nearest neighbors. Although point-based methods demonstrate clear advantages in preserving geometric details, the computational overhead associated with neighborhood construction increases substantially as the point cloud density grows.

## Graph-based methods

Graph-based methods construct the topology and shape of irregular point clouds into graph structures as input (Wang et al., 2019a; Shi and Rajkumar, 2020; Lin et al., 2020; Zhang et al., 2023b). Subsequently, graph neural networks (GNN) perform feature propagation and aggregation over graph structures to learn contextual feature representations for each node. In Simonovsky and Komodakis (2017), 3D points were treated the nodes of graphs, and connections between neighboring points were modeled as edges. An edge-conditioned convolution was then introduced to extract features within fixed-radius neighborhoods. To reduce the data scale and computational complexity of graph construction, some methods pre-segment the point cloud into small, homogeneous regions called superpoints, which serve as nodes for constructing superpoint graphs instead of using individual points (Landrieu and Boussaha, 2019; Hui et al., 2021; Liang et al., 2021; Robert et al., 2023, 2024). For example, Xu et al. (2017) proposed a hierarchical segmentation method that first voxelizes point clouds to reduce complexity and then constructs a superpoint graph based on neighborhood feature similarity. Graph convolution is subsequently conducted at the superpoint level based on edge weights derived from geometric similarity and spatial proximity. Landrieu and Simonovsky (2018) introduced a semantic segmentation framework that combines PointNet for intra-cluster feature extraction with superpoint graphs for contextual aggregation through superedges in large-scale point clouds. Notably, DGCNN (Wang et al., 2019b) is a representative graph-based method that dynamically constructs and updates graphs during training. By incorporating edge convolution modules, it captures both local geometric structures and potential long-range semantic dependencies. Liang et al. (2019) proposed a deep generalized convolution module to capture features and seamlessly propagate them to neighbors while maintaining invariance to point permutations. The integration of depth-wise graph convolution and point-wise convolution in a hierarchical structure ensures the synergy between local and global information. To enhance spatial coverage and reduce theoretical time complexity, Xu et al. (2020) designed a coverage-aware grid query module that leverages the spatial regularity of grids, followed by a graph convolution module named grid context aggregation to integrate context features and coverage information. In Ma et al. (2020), a global context aggregation module based on graph structures was proposed to capture long-range dependencies, combined with an attention-based fusion strategy to adaptively balance local geometric details and global semantic information. Furthermore, Lei et al. (2021) partitioned the spherical space occupied by 3D points into multiple bins linked to learnable parameters and introduced separable spherical convolutional kernels for efficient graph convolutions across large contextual regions. Overall, graph-based approaches can efficiently exploit both local and global contextual information in point clouds via flexible graph structures and convolution operations. However, designing efficient graph construction strategies and convolution operations remains a critical challenge in this research domain.

### 2.3 Attention mechanism in semantic segmentation

The attention mechanism initially achieved groundbreaking success in the field of natural language processing (NLP) (Vaswani, 2017). The core idea is to enable models to dynamically assign attention weights based on the importance of input data, closely resembling the selective attention behavior of the human auditory and visual systems. The transformer architecture, built on the attention mechanism, excels at capturing long-range dependencies (Ramachandran et al., 2019; Cordonnier et al., 2020) and has rapidly become the dominant block of both encoder and decoder designs in pattern recognition tasks (Dosovitskiy et al., 2020; Carion et al., 2020; Wang et al., 2021; Zhu et al., 2021). A standard transformer encoder typically consists of key components such as input embeddings, positional encodings, self-attention modules, normalization layers, feed-forward networks, and skip connections, while the decoder additionally incorporates latent features from the encoder outputs (Vaswani, 2017). Inspired by its success in the NLP domain, the attention mechanism was rapidly introduced into the field of computer vision and initially applied to 2D image processing tasks (Li et al., 2018a; Dosovitskiy et al., 2020; Liu et al., 2021; Xie et al., 2021; Zheng et al., 2021; Cheng et al., 2021a, 2022; Xue et al., 2022; Kirillov et al., 2023), aiming to enhance the model’s ability to capture dependencies across different regions of an image.

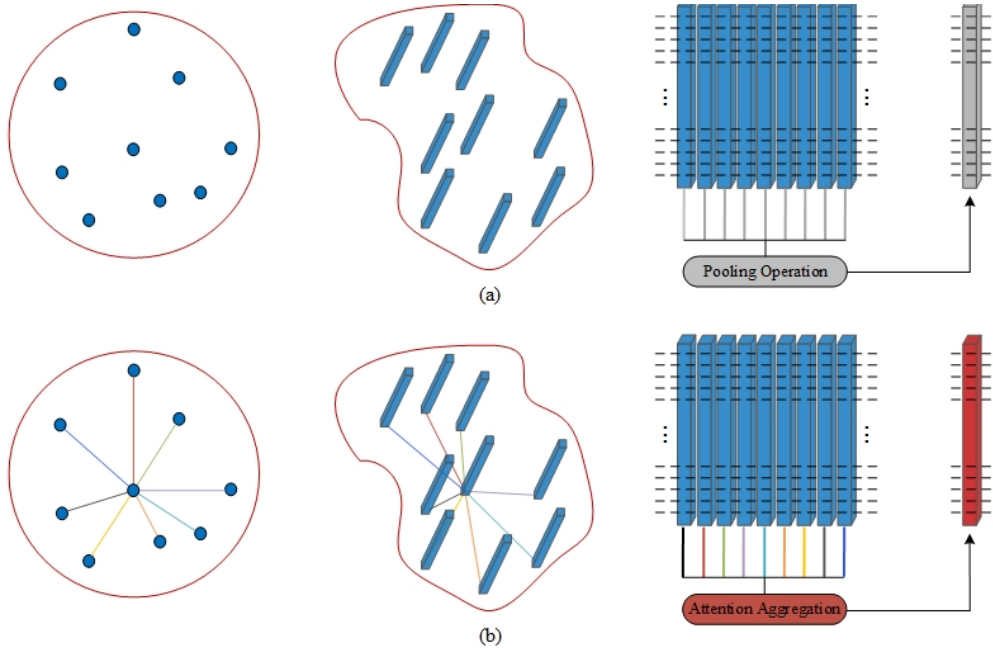


Figure 2.1: Illustration of different feature operations for point cloud processing. (a) presents the pooling in classical convolution. (b) presents the attention-based aggregation.

Compared to the classical convolution, the attention operation provides importance scores based on the relationships between elements, enabling models to focus dynamically on discriminative features or regions and suppress outliers (Naseer et al., 2021; Yang et al., 2021; Khan et al., 2022). This property is particularly beneficial for handling discrete points naturally embedded in 3D space. In point cloud data, local geometric features such as edges, sharp corners, and curved surfaces are typically represented by sparse points, making these features critical for preserving intricate details. However, convolution-based networks commonly rely on pooling operations to reduce computational complexity and aggregate local features. These hard pooling operations focus solely on statistical summaries within local neighborhoods, ignoring the relative positional relationships and topological structures between points, as illustrated in Fig. 2.1(a). Consequently, features in edge regions or geometrically complex areas are often blurred or lost during the pooling process. In contrast, the self-attention mechanism dynamically adjusts feature weights by computing relative positional relationships among points, such as Euclidean distances, normal vector similarities, or feature dependencies, rather than simply taking the maximum or average pooling values, as shown in Fig. 2.1(b). This enables the self-attention mechanism to focus precisely on critical regions and adaptively integrate local features, effectively mitigating the degradation of detailed geometric information inherent to pooling operations.

Given the unordered and irregular nature of point clouds, effectively leveraging the attention mechanism to capture both local and global contextual information remains a major challenge. Projection-based 3D segmentation methods naturally inherit 2D attention modules. For instance, RangViT (Ando et al., 2023) utilized the flexibility of pretrained vision transformers (ViT), transferring large-scale 2D knowledge to LiDAR range images. Similarly, Kong et al. (2023) proposed RangeFormer, which introduced a scalable training strategy by dividing full range images for higher horizontal angular resolution. Huang et al. (2023) leveraged a tri-perspective projection strategy to attain dense 3D occupancy predictions, which is robust to occlusion and sparsity. Several studies have incorporated the attention mechanism into graph structures. In Wang et al. (2019a); Ma et al. (2020); Xie et al. (2020c), attention operations were integrated into graph convolutions to adaptively balance local geometric details and global semantic context. Yue et al. (2022) proposed a dynamic region graph convolution based on the attention mechanism, which adaptively weights features from critical regions while suppressing the influence of noisy points. In addition, some other works have optimized specific modules in previous point-based segmentation networks by leveraging the attention mechanism. Yan et al. (2020) designed an adaptive sampling module enhanced by attention, which reweights features of neighborhood points selected via the FPS algorithm, thereby adjusting sampled point features and mitigating biases caused by outliers. Guo et al. (2021) first introduced a permutation-invariant transformer architecture for point clouds using attention operations to directly process irregular 3D points, focusing mainly on global features.

To enhance local feature aggregation, Zhao et al. (2021) developed a hierarchical architecture constructed on point transformer layers that incorporate trainable positional encodings through the subtraction relation. To better organize local feature sets, Engel et al. (2021) introduced SortNet into a local-global attention framework to capture spatial and higher-order relationships between points across different subspaces. Han et al. (2022) designed a dual-transformer structure that aggregates multi-head self-attention scores along both the point and channel dimensions, learning contextual features across multiple resolutions and scales to enhance feature interactions. Stratified Transformer (Lai et al., 2022) employed a stratified sampling strategy that densely samples nearby points and sparsely samples distant points as keys for each query point to enlarge the effective receptive field and access long-range contexts at a low computational cost. Point-BERT (Yu et al., 2022) combined a self-supervised pretraining paradigm with a transformer architecture. Through masked point modeling and discrete point cloud tokenization, it significantly improved model performance under data-scarce conditions. To solve the problem of significant variation in object sizes, Zhou et al. (2023a) proposed a field-aware transformer, consisting of multi-granularity attention modules and re-attention modules, to customize attention fields for multi-scale objects. To address the high computational cost associated with attention maps, Park et al. (2022) designed a lightweight self-attention module constructed on the voxel-hashing architecture, effectively learning continuous spatial information of 3D points while significantly reducing memory complexity. Robert et al. (2023) introduced a superpoint-based attention strategy that groups local points into superpoints to lower computational costs while maintaining sensitivity to detailed features. Overall, from initially borrowing concepts from NLP to the subsequent development of attention modules and architectures tailored for point cloud data, the attention mechanism has progressively demonstrated its remarkable potential in the field of 3D semantic segmentation by adaptively capturing relationships between points and dynamically focusing on critical feature information.

## 2.4 Semantic segmentation for urban point clouds

With the rapid development of deep learning in the 3D vision domain, the demand for semantic segmentation of large-scale urban point clouds has significantly increased. In a wide range of applications, such as autonomous driving, smart city planning, and infrastructure management (Soilán et al., 2018; Behley et al., 2019; Xu et al., 2021a; Zhou et al., 2023b), accurate semantic annotation of massive point cloud data captured from outdoor urban scenes is of critical importance. However, existing research mainly focuses on indoor point cloud segmentation (Qi et al., 2017b; Zhao et al., 2019; Jiang et al., 2019; Lei et al., 2019; Zhao et al., 2021; Lai et al., 2022), with limited consideration for the unique characteristics of urban environments. Compared to the spatially constrained indoor point clouds, urban point clouds introduce a series of distinct challenges to 3D semantic segmentation due to their inherent properties.

### 2.4.1 Complexity of large-scale urban point clouds

Advancements in 3D scanning technology and integrated acquisition platforms have made it feasible to obtain large-scale, high-density urban point clouds (Tan et al., 2020; Varney et al., 2020; Kölle et al., 2021; Han et al., 2024), which typically contain billions or even more discrete points. Such volume of data, far exceeding that of indoor point cloud datasets, leads to near-exponential growth in memory consumption and computational complexity during processing (Landrieu and Simonovsky, 2018; Thomas et al., 2019). Consequently, conventional deep network architectures often struggle to directly handle these unordered and massive point sets. To address this, Qi et al. (2017b) introduced a hierarchical structure and downsampling-based feature learning scheme to progressively extract semantic features by aggregating points from local point neighborhoods. While this approach reduces complexity, directly applying it to large-scale urban point clouds poses risks of potential loss of critical information during downsampling. Additionally, MinkUNet (Choy et al., 2019) adopted sparse convolutions, and RandLA-Net (Hu et al., 2020) utilized random sampling and local aggregation to improve efficiency in large-scale point cloud processing. Nonetheless, the computational cost of neighborhood search and convolution operations remains substantial.

Another prominent characteristic of urban point clouds is their extreme scene diversity. Different cities, districts, or even the same region at different times can exhibit highly variable architectural styles, vegetation types, and road layouts. This urban complexity requires models not only to capture intricate contextual dependencies but also to possess cross-scene generalization capabilities. However, achieving such generalization heavily relies on access to large-scale annotated training datasets (Li et al., 2020). Despite the availability of several urban point cloud datasets, their scale and diversity remain limited due to the high cost of manual labeling. To alleviate this, some studies have explored weakly supervised and self-supervised methods to enhance the cross-domain generalization using limited annotations (Xie et al., 2020a; Wang and Yao, 2022; Min et al., 2023). Moreover, cross-platform data discrepancies introduce additional challenges for deep learning-based point cloud segmentation. Large-scale urban point clouds are typically acquired via various integrated platforms, including mobile laser scanning (MLS), terrestrial laser scanning (TLS), and airborne laser scanning (ALS). However, differences in sensor mounting height, laser scanning patterns, and movement trajectories result in significant variations in point density and distribution (Xie et al., 2020b; Huang et al., 2021). For instance, TLS data offers high precision and point density but has limited coverage with highly uneven spatial distributions; ALS data provides complete city-level coverage but are primarily concentrated on horizontal surfaces such as rooftops and tree canopies; MLS data excels at describing detailed street structures like roads, facades, and municipal facilities, yet exhibits sparse coverage of building rooftops. Therefore, models trained on data from specific platforms often struggle to generalize directly to data from other platforms. To

bridge the gap, some research has focused on domain adaptation techniques (Yi et al., 2021; Saltori et al., 2023; Xiao et al., 2023; Chen et al., 2024), aiming to align distributions and learn invariant representations across different data domains. Although most existing studies reported high performance when training and testing on the same domain, they often lack comprehensive evaluations of generalization across both cities and sensing platforms. These evaluations demonstrate that the model’s robustness in learning semantic features that generalize across cities and platforms by avoiding overfitting to region- or platform-specific attributes. Datasets like WHU-Urban3D (Han et al., 2024) enable such generalization evaluations by offering the unified annotation standard across distinct urban areas and platforms. Nevertheless, comprehensive multi-city datasets with standardized labeling and diverse sensing platforms remain scarce. Addressing this gap is a critical direction for future research.

#### **2.4.2 Fine-grained and imbalanced classes in urban point clouds**

High-density point clouds captured by advanced platforms reveal fine-grained geometric details in urban scenes, while driving an increasing demand for finer boundary distinction between fine-grained semantic categories (Zhang et al., 2021). Some studies, such as the hierarchical point-edge interaction network proposed in Jiang et al. (2019) and the graph attention convolution introduced in Wang et al. (2019a), have made significant progress in preserving boundary details. Furthermore, the boundary-aware segmentation framework developed in Jung et al. (2021) has further improved the accuracy of building boundaries in urban scenes. Nevertheless, most methods still struggle to classify fine-grained boundaries of similar geometric structures, such as earth roads, paved roads, and low vegetation, subtle architectural elements like chimneys and rooftops, as well as traffic cones and barriers (Huang et al., 2021). This limitation is often attributed to the restricted receptive fields and the gradual loss of fine-grained information in deep network layers. To alleviate this issue, Zhang et al. (2022a) incorporated original local surface features into multiple stages of the attention network to enhance the fine-grained segmentation performance. Furthermore, objects in urban environments exhibit significant size variations, ranging from large structures such as buildings and bridges to small objects like traffic signs and pedestrians. For small-scale objects, low sampling rates may result in the loss of critical features during the downsampling process. Conversely, for large-scale objects, aggregating widely distributed local features to achieve globally consistent semantic labeling remains a major challenge (Huang et al., 2019, 2020). Single-scale feature extraction networks can not effectively manage such scale variations. Thus, multi-scale feature fusion strategies, including the multi-scale grouping and the multi-resolution grouping, employed in Qi et al. (2017b); Liu et al. (2019a) leverage convolutional kernels or receptive fields at different scales for feature extraction. Thomas et al. (2019) defined a set of deformable kernels of different sizes, adapting to objects exhibiting diverse scales and geometric shapes. Other studies

have addressed this issue by directly leveraging the structural characteristics of objects. Zhang et al. (2022b) explored a two-stage approach to tackle multi-scale category segmentation in large-scale urban point clouds, which first detects regions potentially containing objects of different scales and then performs refined segmentation. However, both multi-scale fusion strategies and two-stage processing approaches still face challenges in terms of improving computational efficiency.

In urban point clouds, variations in object size and quantity often leads to the severe class imbalance issue. For instance, roads and buildings occupy the majority of points, whereas small objects such as traffic lights and utility poles account for only a tiny fraction, which are critical for practical applications (Pan et al., 2020; Bayrak et al., 2024; Duan et al., 2024). When dealing with non-uniformly distributed point clouds, small objects in sparse regions often lack sufficient structural features, making the segmentation of rare classes even more challenging (Zhang et al., 2022a). During training, this class imbalance biases models toward majority classes with abundant samples, while underfitting rare classes, resulting in reduced segmentation accuracy for rare categories during inference (Zhang et al., 2025). Different strategies have been proposed to address this issue. Approaches based on data distribution include oversampling rare classes, undersampling majority classes, and augmenting rare class data to alleviate imbalance. Griffiths and Boehm (2019) proposed a weighted point cloud augmentation strategy by replicating minority-class points to alleviate imbalance. Ren and Xia (2023) further designed a stack autoencoder-based oversampling strategy that synthesizes new 3D points by learning the distribution of underrepresented classes. Other works, such as Fang et al. (2021); Xu et al. (2021a); Xiao et al. (2022), introduced synthetic rendering or instance-level augmentation, injecting rare-object point cloud fragments into training data to explicitly increase the occurrence of rare classes. Approaches based on loss function adjust the loss computation by assigning higher weights to rare samples, guiding the model to focus more on them during training (Eigen and Fergus, 2015; Lin et al., 2017; Ren et al., 2020). Lin and Nguyen (2020) proposed a two-stage framework that combines adaptive loss reweighting with mixed sampling to enhance rare class learning. Approaches based on network design aim to develop customized architectures by incorporating distinctive features of rare classes to address class imbalance. For instance, Zhang et al. (2023c) introduced a target-guided learning framework, which injects independent structural features of individual objects to focus on rare class segmentation. Building on this, Zhang et al. (2025) developed a target-aware aggregator to refine attention scores, specifically addressing rare class segmentation in large-scale urban point clouds. Despite these efforts, effectively improving segmentation performance of rare classes continues to present an ongoing challenge in complex, large-scale urban point cloud scenarios.

In summary, the semantic segmentation of large-scale urban point clouds is still regarded as a highly challenging research area due to the unique characteristics of urban environments. Although numerous high-performing segmentation methods for point clouds have emerged in recent years,

achieving robust, accurate, and efficient semantic segmentation of complex, large-scale urban point clouds, particularly for fine-grained and rare classes, remains a key focus for future research.

## Chapter 3

# Methodology

To tackle the challenges of semantic segmentation in large-scale point clouds, particularly the recognition and segmentation of rare classes, we propose an integrated framework consisting of various interconnected components. Section 3.1 provides a detailed discussion of the attention-based network for large-scale point clouds, which is central to our framework, enabling the model to focus on important features across massive datasets. To address the challenges of the limitations of weak local representations, surface context embedding is presented in Section 3.2, where we explore how contextual information derived from surface features can improve fine-grained semantic segmentation. Recognizing the scarcity of data for rare classes, Section 3.3 introduces a context-guided augmentation strategy tailored for rare classes, which enhances the diversity under constrained scenarios for underrepresented categories. After that, we shift focus to the segmentation model for rare classes, exploring how geometric and topological features at the individual level can be leveraged to improve segmentation outcomes. This is achieved through the target-aware attention mechanism described in Section 3.4, which adaptively modulates attention weights based on target regions to boost segmentation performance for rare classes. Together, these components address key issues such as the irregularity and sparsity of point clouds, class imbalance, the difficulty of representing rare class features, and the need for adaptive attention to rare and complex objects in large-scale scenes.

### 3.1 Attention network for large-scale point clouds

In this section, we present a comprehensive description of the proposed attention-based network framework, specifically designed for semantic segmentation tasks in large-scale point clouds. By

incorporating an advanced attention mechanism, the network adaptively prioritizes key regions within the scene while efficiently extracting multi-scale information.

Key modules of the proposed framework include the self-attention operator for the local feature extraction (Section 3.1.1), the multi-head attention mechanism to enhance feature representation by capturing dependencies in multiple feature subspaces (Section 3.1.3), and up-sampling and down-sampling layers to enable hierarchical feature refinement (Section 3.1.4). Additionally, skip connections facilitate the seamless transfer of information across layers, preserving critical semantic details from earlier stages of the network, while the multi-level network structure enables progressive feature aggregation at different spatial scales (Section 3.1.4). These modules work collectively to form a tightly integrated network framework that not only ensures effective handling of irregular and sparse point cloud data but also enhances the network’s capability to capture global context and intricate structural details of local regions. This design provides a robust foundation for addressing the complexities of large-scale point cloud semantic segmentation.

### 3.1.1 Local self-attention operator

As part and parcel of the proposed attention network, the local self-attention operator plays a pivotal role in capturing local dependencies between points. Given that the point cloud  $X$  consists of unordered sets of points embedded in 3D space, and the self-attention operator is inherently permutation-invariant for input elements, the proposed attention-based segmentation network is entirely constructed using the self-attention mechanism and point-wise operations. The designed local self-attention operator is composed of four key components: spatial attention encoding, redundant positional encoding, multi-head splitting and local attention aggregation. Fig. 3.1 provides a detailed illustration of the integration of the above components to capture the correlation between 3D point elements considering local dependencies.

To extract the local features of a given input point  $x_i$ , the spatial distribution of its  $k$ -nearest neighbors (KNN) point set  $X_i \in X$  is encoded. Each point  $x_j \in X_i$  is treated as an adjacent neighbor of the input point  $x_i$ , contributing to the representation of the local geometry. Following the method proposed in Dosovitskiy et al. (2020), a linear mapping is applied to project the input features into embedding vectors suitable for the attention module, facilitating dimensional transformation and enabling effective feature interaction. In our framework, the term for vector embedding of a point can be defined as:

$$\begin{aligned} q_i^l &= \tau(f^l, W_q), \\ k_j^l &= \tau(f^l, W_k), \\ v_j^l &= \tau(f^l, W_v), \end{aligned} \tag{3.1}$$

where  $f^l$  represents attributes or intermediate learned features for the  $l_{th}$  layer of the corresponding point, while  $\tau(\cdot)$  is the embedding function implemented through one simple linear layer. The values  $q_i^l$ ,  $k_j^l$  and  $v_j^l \in \mathbb{R}^D$  denote different token vectors of feature dimension  $D$  mapped by learnable transformation matrices  $W_q$ ,  $W_k$  and  $W_v$ .

To establish local relationships among points in the attention module, the subtraction relation is applied to calculate the local correlation for point  $x_i$ , with encoded positional information integrated to enrich spatial awareness. This is expressed as follows:

$$c_{i,j}^l = \alpha \left( \beta \left( q_i^l - k_j^l + p_{i,j}^l \right) \right) \quad (3.2)$$

where  $c_{i,j}^l$  describes the similarity between points  $x_i$  and  $x_j$ . The value  $q_i^l$  is the query vector of point  $x_i$ , while  $k_j^l$  is the key vector of point  $x_j \in X_i$ . The position encoding function  $p_{i,j}^l$  aims to encode the relative spatial information between local points, which is introduced in Section 3.1.2. The relation mapping function  $\beta(\cdot)$  produces correlation vectors for feature aggregation, which is implemented by a multilayer perceptron (MLP), i.e., two batch normalization layers and linear layers followed by rectified linear unit (ReLU) (Glorot et al., 2011) layers. The normalization function  $\alpha(\cdot)$  is implemented by softmax.

Before calculating the attention score of point  $x_i$ , in addition to the relation function, we also enhance the value part of attention module by incorporating encoded spatial information, as described below:

$$\nu_j^l = v_j^l + p_{i,j}^l \quad (3.3)$$

where  $\nu_j^l$  is the enhanced value vector of point  $x_j \in X_i$ , which is combined with the positional encoding  $p_{i,j}^l$  using a lightweight element-wise addition operation. This integration process for the value vector  $v_j^l$  leverages the point's local original information effectively, while introducing only a minimal increase in computational overhead.

Building on the above enriched value representation, the point-wise attention score between two points is calculated to quantify the influence of a neighboring point on the target point within the local context. This computation is expressed as follows:

$$a_{i,j}^l = c_{i,j}^l \odot \nu_j^l \quad (3.4)$$

where  $a_{i,j}^l$  represents the attention score of the input point  $x_i$  for its neighboring point  $x_j$ . The value  $c_{i,j}^l$  is the correlation vector between points  $x_i$  and  $x_j$ , and the vector  $\nu_j^l$  serves as the input value source for the attention module. The symbol  $\odot$  denotes self-attention operation using dot

product, namely the element-wise multiplication, which plays the role of attention aggregation as illustrated in Fig. 3.1.

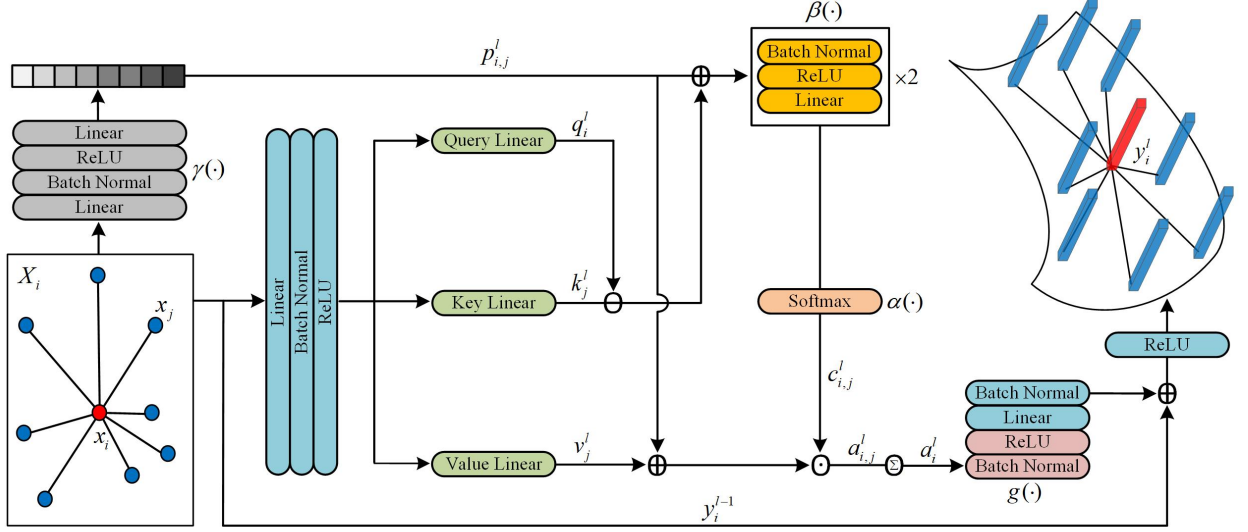


Figure 3.1: Schematic diagram of the local attention aggregation module. The red dot represents the selected target point, and blue dots represent its neighboring pointset. Note that this is only an example of a single point, and indeed all target points will undergo the same operation.

To further aggregate the information provided by the entire local scope, the attention aggregation step combines the individual point-wise attention scores from the neighboring points into a unified representation for the target point  $x_i$ . Thus, the local attention aggregation can be formulated as:

$$a_i^l = \sum_{x_j \in X_i} a_{i,j}^l \quad (3.5)$$

where  $a_i^l$  stands for output attention score of the input point  $x_i$  in a local scope. This is implemented through a summation operation over its neighborhood  $X_i$ . Notably, in the single-head attention module, the attention score  $a_i^l$  is directly passed through a batch normalization layer followed by a ReLU activation layer, represented as follows:

$$y_i^l = g(a_i^l) + y_i^{l-1} \quad (3.6)$$

where the output  $y_i^l$  represents the final attention score of the input point  $x_i$  within its local region. To ensure the robustness of training, a residual connection is introduced by adding the value  $y_i^{l-1}$  to the output, following the approach outlined in (He et al., 2016). The residual connection and the local attention operation together form our local attention aggregation module. In contrast to hard aggregation operations such as max or mean pooling commonly used in convolutional networks, our

attention-based operator thoroughly preserves and inherits the local features from different layers in high-dimensional space.

### 3.1.2 Redundant positional encoding

In the attention mechanism, injecting some information about the relative position of the tokens is essential for leveraging the order of the input sequence (Vaswani, 2017). To achieve this, positional encoding is typically added as auxiliary information to the input features, enhancing the model’s ability to interpret sequential and positional relationships. Similarly, in attention-based 3D segmentation frameworks, the attention mechanism captures feature interactions by computing correlations between points. While this mechanism adaptively aggregates neighborhood information, its inherent permutation invariance (i.e., insensitivity to the order or position of the input data) can introduce challenges. Specifically, in point cloud data, the relative spatial distribution of points is a key characteristic of the geometric structure. Simply calculating attention weights without incorporating positional information can result in segmentation models lacking spatial awareness.

For the segmentation task of point clouds, the semantic information of each point is inherently dependent on its spatial location and its relationships within the overall structure. Therefore, positional encoding must be introduced into the point cloud data to enhance the attention module’s ability to perceive relative spatial features. Such positional embedding not only enables the model to better capture local shapes by injecting positional information but also aids in understanding the spatial distribution of points within the global scene, which is particularly critical for processing large-scale point clouds. Regarding sequential and image grid data, absolute positional encoding schemes such as sine and cosine functions are commonly employed, as they perform well on regularly distributed data. In the case of irregular 3D point cloud data, the 3D coordinates of points naturally serve as candidates for positional encoding, which directly describe the location information of points in space. However, only adopting relative coordinates in the cartesian coordinate system as positional encoding can be insufficient for capturing contextual spatial relationships and local geometric dependencies (Hu et al., 2020; Fan et al., 2021). Different from Zhao et al. (2021), we propose a learnable redundant positional encoding  $p_{i,j}^l \in \mathbb{R}^{k*c}$  to point cloud data. The design is as follows:

$$p_{i,j}^l = \gamma \left( (\zeta_i - \zeta_j) \textcircled{\text{C}} \|\zeta_i - \zeta_j\| \textcircled{\text{C}} (\theta_i - \theta_j) \textcircled{\text{C}} (\phi_i - \phi_j) \right) \quad (3.7)$$

where  $\textcircled{\text{C}}$  is the concatenation operation,  $\zeta_i$  and  $\zeta_j$  are respectively the 3D coordinates of points  $x_i$  and  $x_j \in X_i$ .  $\zeta_i - \zeta_j$  is relative position vector and  $\|\zeta_i - \zeta_j\|$  is euclidean distance of neighboring points. The angles  $\theta$  and  $\phi$  denote the azimuthal and polar angles in the spherical coordinate system, respectively. To address the inherent periodicity of angular values, each angle is encoded as a two-dimensional vector using its sine and cosine components, thereby avoiding discontinuities

near angular boundaries. The symbol  $\gamma(\cdot)$  is the encoding function consisting of two linear layers, one batch normalization layer and one ReLU layer.

By introducing a trainable encoding function, this strategy merges the relative positions and geometric relationships of point clouds into rich positional representations, as illustrated in Fig. 3.1. On the one hand, the redundant encoding scheme enhances the representation capability of the attention module. On the other hand, the trainable positional encoding adapts to different scenes by optimizing the learnable parameters. For instance, in sparse point cloud regions, the encoding function can prioritize geometric distances, whereas in dense regions, it may rely more on relative positional features. This adaptive mechanism ensures that the attention module remains robust when handling complex point cloud scenarios.

### 3.1.3 Multi-head attention

Multi-head attention is extended method based on the attention mechanism, enabling the model to jointly focus on information from various representation subspaces at different positions (Vaswani, 2017). Compared to the single-head attention, the multi-head structure leverages multiple independent attention heads to capture diverse feature representations, demonstrating significant advantages in complex point cloud segmentation tasks involving multi-scale targets.

Firstly, the multi-head attention facilitates the effective combination of global and local features. Different attention heads can independently focus on long-range dependencies and local neighborhood characteristics, allowing the model to highlight multi-scale target features while preserving global consistency. Such a capability is particularly beneficial for capturing multi-scale contextual information within point clouds, enhancing feature representation in complex scenarios. Secondly, through parallel computation, the multi-head attention provides diverse representations of complex geometric structures. Different heads can independently capture different geometric features within the same region of a point cloud. For instance, in building and road segmentation tasks, some attention heads may concentrate on the overall contours of buildings, while others focus on local detail features such as corner points or road boundary information. Lastly, by distributing the computation of attention weights across multiple heads, the multi-head structure improves model stability and robustness. The single-head attention can exhibit excessive sensitivity to noisy or outlier points during training. In contrast, multi-head attention balances weight distribution across multiple subspaces, reducing the model’s reliance on noisy points.

The core idea of this approach is to divide the input features into multiple subspaces, compute attention in parallel across these subspaces, and then aggregate the feature information. The formula is expressed as follows:

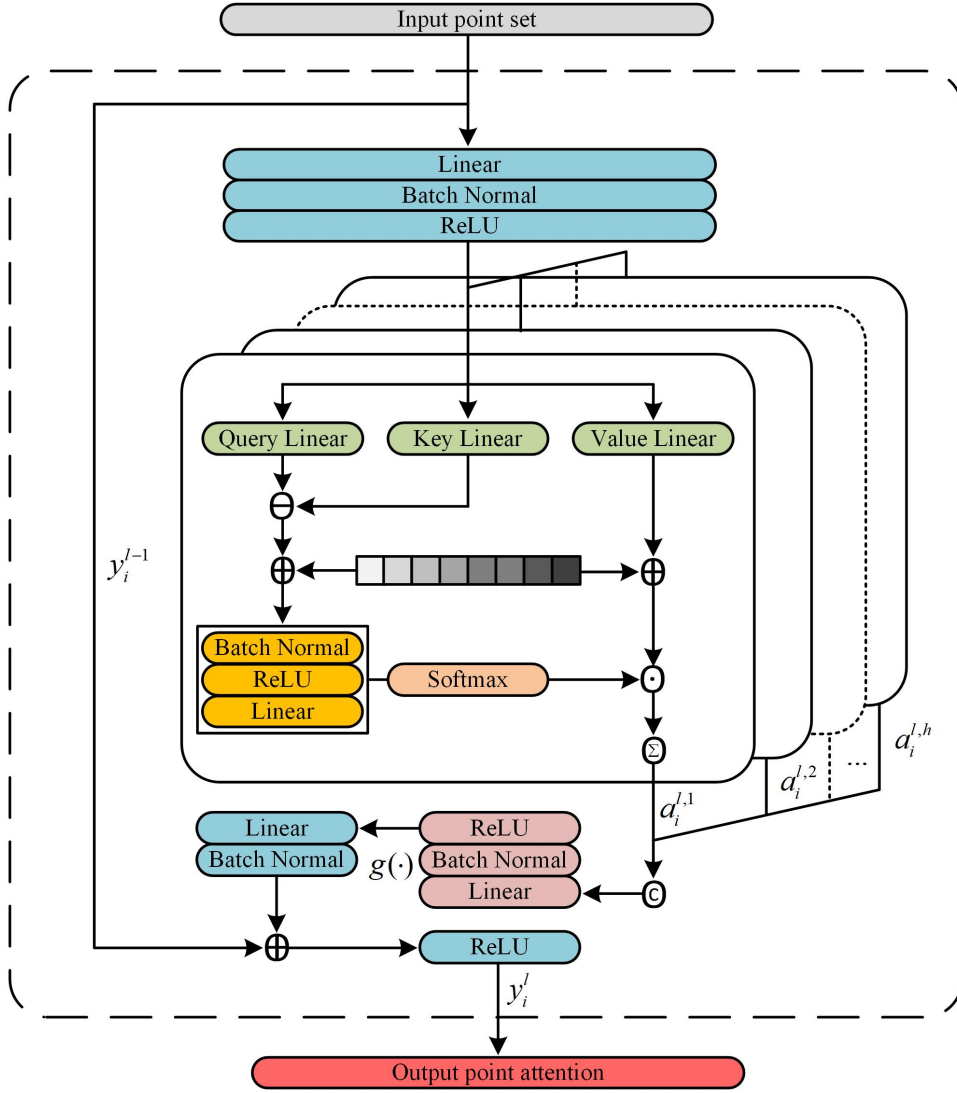


Figure 3.2: Illustration of the multi-head attention. Multiple parallel heads independently perform local attention aggregation in different representation subspaces, and their outputs are concatenated to produce the final point features.

$$y_i^l = g\left(a_i^{l,1} \odot a_i^{l,2} \odot \dots \odot a_i^{l,h}\right) + y_i^{l-1} \quad (3.8)$$

where the output  $y_i^l$  represents the final multi-head attention score of the input point  $x_i$  within its local region. This score is derived from the division of the query, key and value in Equation 3.1 into different subspaces. Specifically, the vectors  $q_i^l$ ,  $k_j^l$  and  $v_j^l \in \mathbb{R}^D$  are projected along the channel dimension using  $h$  distinct linear transformations, mapping them from the full dimension  $D$  into subspaces of dimension  $d$ , where  $h$  represents the total number of attention heads. Since the

dimension of each subspace is  $d = D/h$ , the overall computational overhead remains comparable to that of the single-head attention with the full dimension  $D$ . In parallel, multiple attention heads independently follow the workflow of the self-attention operator described in Section 3.1.1 to aggregate attention scores from the local region, denoted as  $a_i^{l,h}$ . Subsequently, the outputs from all attention heads are concatenated, represented by the operation  $\odot$ . This combined representation is then passed through the mapping function  $g(\cdot)$ , which formed by one linear layer and one batch normalization layer. To enhance training robustness, the value  $y_i^{l-1}$  is added as a residual to the output (He et al., 2016). Fig. 3.2 shows the structure of the multi-head attention. The aforementioned operations will be executed for each individual point, producing point-wise attention scores  $A \in \mathbb{R}^{N \times D}$  for entire point clouds.

### 3.1.4 Hierarchical network architecture

Building on the local attention operator, we have developed a network architecture tailored for point cloud segmentation tasks, as illustrated in Fig. 3.3. By incorporating sampling operations, a hierarchical structure, and skip connection mechanisms, the architecture effectively addresses the challenges of the irregularity and multi-scale characteristics of point cloud data. The overall design aims to fully capture local geometric features and global contextual information, while maintaining robust adaptability to complex scenes and multi-scale targets.

#### Sampling operation

In point cloud segmentation networks, where data typically consists of a large number of points, sampling operations serve as one of the foundational modules that balance computational complexity and geometric integrity, comprising both down-sampling and up-sampling processes. Through iterative downsampling, high-resolution point cloud features are gradually compressed into a low-resolution, high-dimensional feature space to extract global semantic information. Conversely, up-sampling restores the low-resolution features back to high-resolution space, recovering details to produce precise segmentation results.

The purpose of down-sampling is to reduce the number of points while maintaining good coverage and distribution of the point cloud. We adopt the farthest point sampling (FPS) strategy (Qi et al., 2017b), which ensures that the sampled point clouds achieve spatial uniformity while preserving critical geometric structure features. As outlined in Algorithm 1, this method iteratively selects the point farthest from the already-sampled set. Specifically, given a set of 3D candidate points denoted as  $P$  and the target number of samples  $M$ , the algorithm initializes the sampling set  $S$  with a randomly selected seed point  $p_r$ . This seed point is inserted into  $S$  and removed from the candidate set. During each iteration, the algorithm computes the minimum distance  $d_i$  between

each remaining candidate point and the points in the sampling set  $S$ . The point  $p^*$  that maximizes  $d_i$  is then selected and added to  $S$ . This iterative process continues until the sampling set contains the desired  $M$  points, at which point the algorithm returns the final sampling set  $S$ . Compared to random sampling, the FPS algorithm better preserves geometric information, making it particularly suitable for capturing local features in complex scenes. On the other hand, up-sampling aims to map low-resolution point cloud features back to high-resolution point clouds, recovering detail and ensuring accurate segmentation boundaries. We utilize the trilinear interpolation method, which efficiently reconstructs the features of target points based on their relative distances to the neighboring points. Additionally, before each up-sampling operation and after each down-sampling operation, an MLP composed of a linear layer, a batch normalization layer, and a ReLU activation layer is incorporated. Together with the sampling operation, this forms a sampling module that embeds features for the subsequent blocks.

---

**Algorithm 1** Farthest point sampling for 3D point clouds

---

**Input**

- A set of 3D candidate points  $P = \{p_i\}_{i=1}^N$
- The target number of samples  $M$

```

1: Initialize the sampling set  $S \leftarrow \emptyset$ 
2: Randomly select one point  $p_r$  from  $P$  as the initial sampled point
3: Insert  $p_r$  into the sampling set:  $S \leftarrow S \cup \{p_r\}$ 
4: Remove  $p_r$  from the set of points:  $P \leftarrow P \setminus \{p_r\}$ 
5: for  $k \leftarrow 2$  to  $M$  do
6:   for each point  $p_i$  in the set  $P$  do
7:     Calculate the closest distance from  $p_i$  to  $S$ :  $d_i = \min_{p_s \in S} \|p_i - p_s\|$ 
8:   end for
9:   Select the point  $p^*$  that maximizes  $d_i$ :  $p^* = \arg \max_{p_i \in P} \{d_i\}$ 
10:  Update the sampling set:  $S \leftarrow S \cup \{p^*\}$ 
11:  Remove  $p^*$  from the candidate set:  $P \leftarrow P \setminus \{p^*\}$ 
12: end for
13: return  $S$ 

```

**Output**

- The final sampling set of  $M$  points
- 

## Hierarchical structure

The network adopts a symmetric encoder-decoder architecture inspired by the design of U-Net (Ronneberger et al., 2015), aiming to achieve high-precision segmentation of point clouds through

progressive feature extraction and reconstruction. The encoder progressively maps high-resolution point cloud features into a low-resolution, high-dimensional feature space through successive down-sampling modules and local attention aggregation modules. Conversely, the decoder reconstructs these low-resolution features back into high-resolution space via up-sampling modules and local attention aggregation modules, enabling fine-grained segmentation of the point cloud. This encoder-decoder structure efficiently integrates local geometric information with global contextual relationships in the point cloud, ensuring that features from sparse regions and small objects are not overlooked. Notably, the multi-level design allows flexible adjustment of sampling rates and hierarchical layers to accommodate various scales of point cloud data.

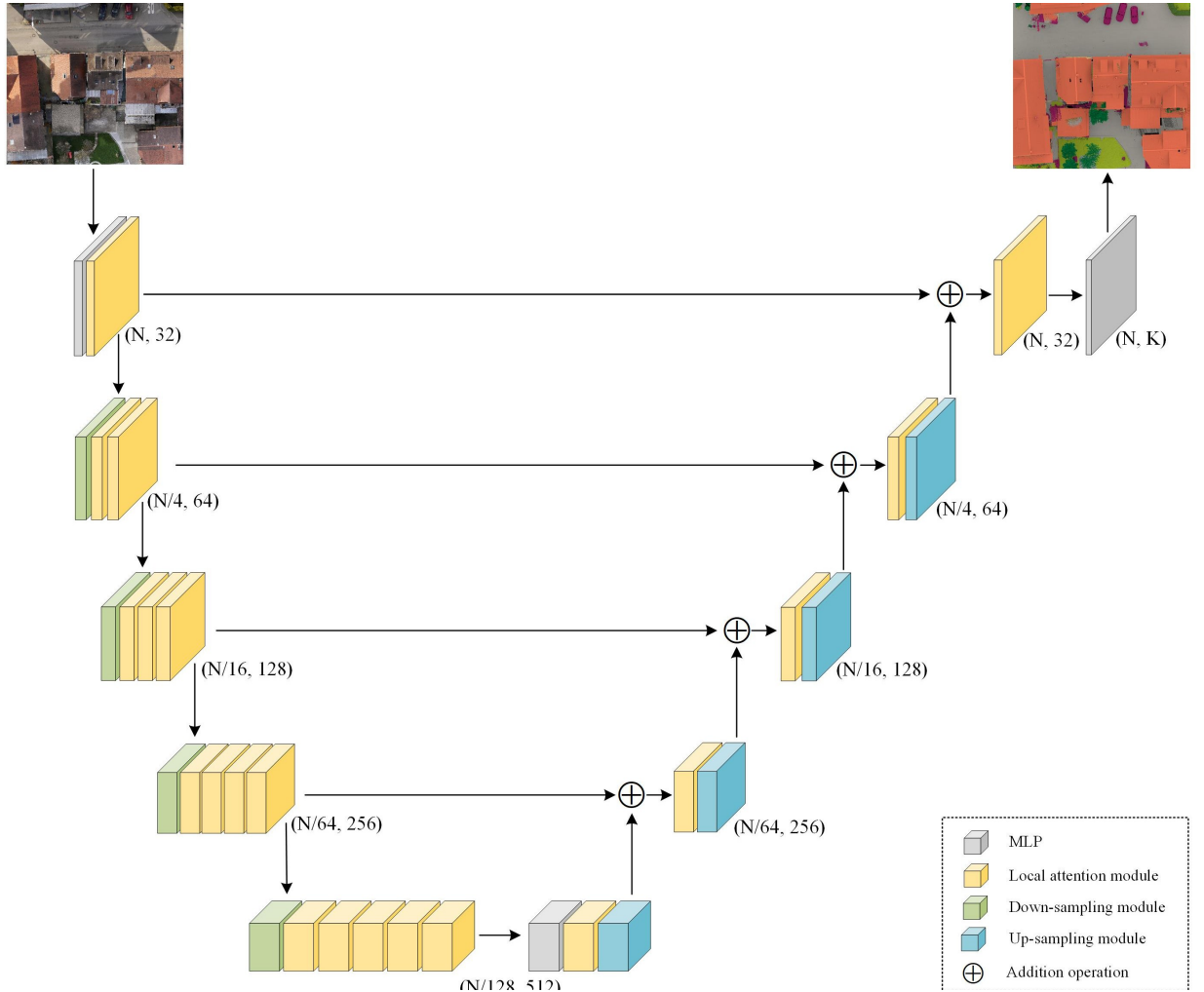


Figure 3.3: Detailed architecture of the hierarchical attention network. Each local attention module learns point-wise features by attending to spatially nearby points, while the U-Net-like structure progressively aggregates global context for point cloud segmentation.

As illustrated in Fig. 3.3, our network consists of four encoder layers and four decoder layers. At each sampling step, we adopt a fourfold decimation ratio. Only 25% of the attention scores are retained after each down-sampling operation, whereas they are restored to four times their resolution following each up-sampling operation. Hence, the combination of down-sampling and up-sampling modules with local attention aggregation modules in the hierarchical network architecture establishes a robust foundation for subsequent feature extraction and multi-scale fusion in large-scale point clouds.

### Skip connection

In the hierarchical network structure, repeated up-sampling and down-sampling can lead to significant information degradation, limiting the expressive capacity of point cloud features. Skip connections (Ronneberger et al., 2015) are introduced to address this issue by facilitating the direct transfer of information across layers. Specifically, skip connections deliver the local geometric features extracted from shallow layers of the encoder directly to the corresponding deeper layers of the decoder. Before performing the skip connection operation, the features to be connected are first processed individually through MLPs. This workflow forms the skip connection module. As depicted in Fig. 3.3, the local attention features from each encoder layer are combined with the corresponding decoder layer's features through skip connections using an element-wise addition operation. By combining local spatial disparities with global consistency, this integration strengthens the discriminative ability of point-wise features for detailed distinctions, particularly improving the representation of complex boundaries and intricate regions.

The overall network architecture can effectively capture long-range dependencies and complex interactions between points in large-scale point clouds. For the semantic segmentation task, the proposed attention network should have an output label for each point. The 4-by-4 symmetrical structure can guarantee the exact correspondence between the input point features and their respective output labels. In the final layer of the network, an MLP is applied to map the point-wise features to the label space. All learnable parameters of the network are then updated by optimizing the cross-entropy loss function:

$$loss_{seg} = - \sum_{i=1}^N \sum_{j=1}^K \left( y_{ij} \log(p_{ij}) + (1 - y_{ij})(1 - \log(p_{ij})) \right) \quad (3.9)$$

where  $N$  and  $K$  represent the total number of points and the total number of classes, respectively. The predicted probability that the  $i_{th}$  point is classified into the  $j_{th}$  class is expressed as  $p_{ij}$ , while  $y_{ij} \in [0, 1]$  serves as the ground-truth label, indicating whether the  $i_{th}$  point belongs to the  $j_{th}$  class.

## 3.2 Local context embedding for fine-grained segmentation

The hierarchical network structure in Section 3.1, combined with sampling operations, significantly improves the efficiency and capacity of the model in handling large-scale point cloud data. However, the down-sampling process inevitably leads to the loss of original local information. In particular, critical local structural details, such as boundary points, regions with curvature variations, and the geometric features of small objects, can be weakened due to the reduced resolution. This loss of local information poses considerable challenges for fine-grained segmentation tasks, especially in 3D scene understanding of complex large-scale point clouds, where insufficient capture of intricate local geometric details becomes a critical limitation. To address this issue, we introduce a local context embedding method that enables the hierarchical network to effectively preserve and leverage variations in original local contextual information during learning. Specifically, spatial relationships between neighboring points are extracted as surface features and subsequently incorporated directly into point-wise feature representations, which serve as inputs to the backbone network. By embedding these localized details into the feature extraction pipeline, the method enhances the network’s expressive capability for the subtle variations in local spatial characteristics, while bridging the gap between global structure learning and local feature preservation.

### 3.2.1 Local surface features

Local spatial information characterizes the geometric distribution properties of each point within its neighborhood, providing a structural representation of the approximate surface in a local region, as illustrated in Fig. 3.4. Such surface information derived from adjacent relationships can capture the irregularities and fine details of the local geometry, particularly in complex point cloud scenarios such as boundary regions. To construct the local surface feature for each point  $p$ , only descriptors with strong semantic interpretation are selected, which are defined as one 6-tuple as follows:

$$f_p = (n_{px}, n_{py}, n_{pz}, c_p, r_p, s_p) \quad (3.10)$$

where  $n_{px}, n_{py}, n_{pz}$  represent components of the normal vector along the  $x$ ,  $y$ , and  $z$  axes, characterizing the orientation of the surface at a specific point. These normals are critical for distinguishing between flat and inclined surfaces in a local region. The value  $c_p$  quantifies the change of surface curvature, which provides a measure of how sharply the surface bends at point  $p$  and reflects the degree of local surface variation. The parameter  $r_p$  is the mean residual distance from the point  $p$  to its fitted surface, which stands for the roughness of the local surface. The value  $s_p$  denotes the sphericity to evaluate how closely the local surface resembles a sphere. Together,

these parameters offer a comprehensive description of surface geometry, which enables detailed analysis for the local perception of the network.

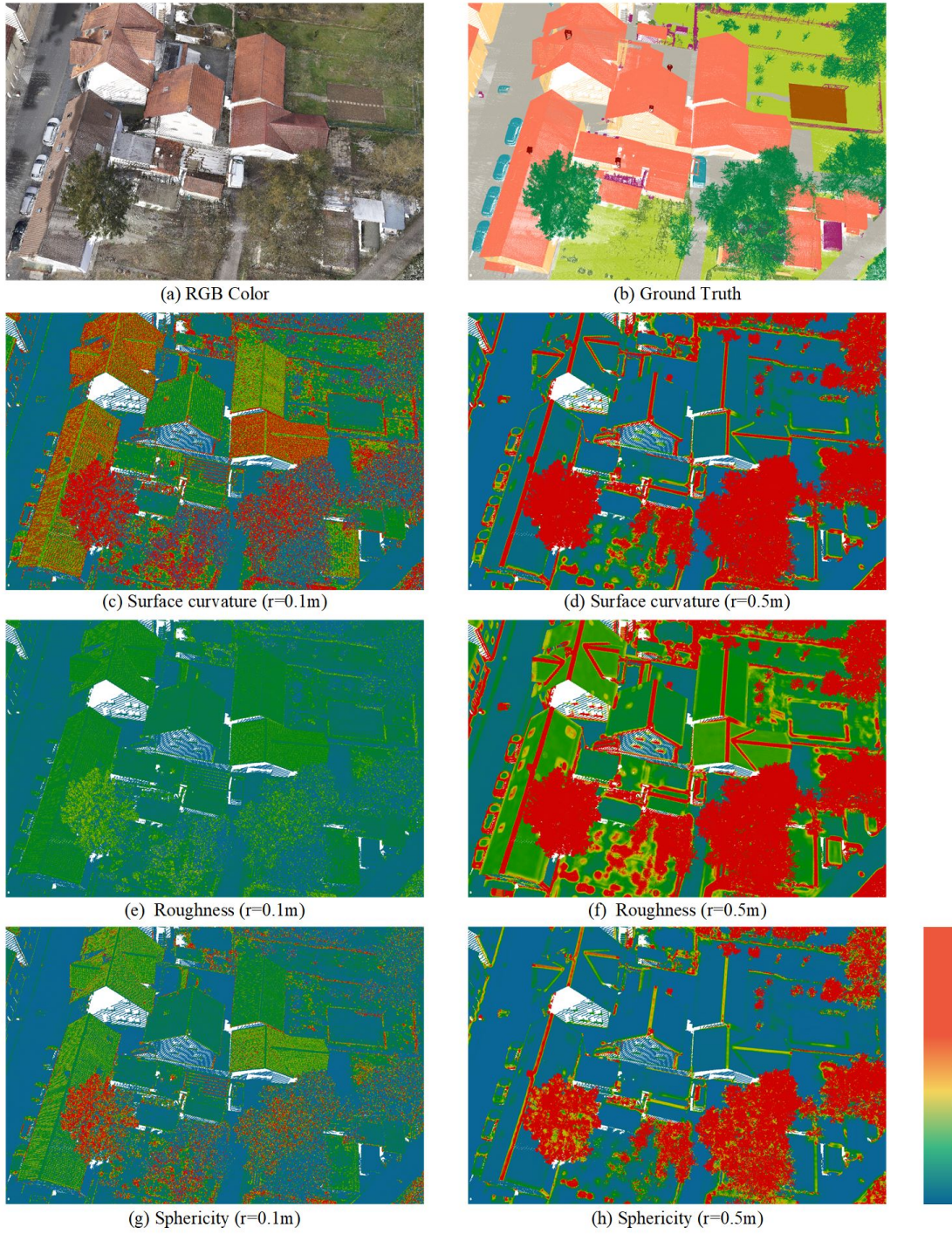


Figure 3.4: Illustration of surface descriptors at different scales. Features extracted at smaller scales capture local geometric details, while larger scales encode broader contextual information.

In order to generate a meaningful description of the local surface properties of a single 3D point, it is essential to establish its adjacent relationship based on spatial distance. However, the KNN method lacks adaptability when handling point clouds with varying densities. In dense regions, the KNN-defined neighborhood may be confined to an overly small area, resulting in incomplete geometric information. Conversely, in sparse regions, the KNN method may include points that are too distant, diluting the representation of local geometric relationships. In contrast, the radius-based ball query method defines neighborhoods using a fixed radius around each 3D point, allowing the number of neighboring points to adjust dynamically. This method better accommodates variations in point cloud density and local geometric characteristics, while maintaining consistency in the spatial relationships of neighborhood points. Therefore, we adopt the radius-based ball query method to define neighborhoods for each point in this work.

Since the goal is to enhance the network’s capacity to identify fine-grained details, we tend to adopt relatively small values of radius, which enables the effective capture of geometric details and subtle variations in local structures crucial for accurate segmentation. While the previous research (Weinmann et al., 2015) has demonstrated that the dynamically defined neighborhoods based on local geometry can significantly improve segmentation results in scenarios with varying point densities, such approaches come with high computational costs for large-scale, high-density point cloud data. As a more computationally efficient alternative, the constant neighborhood definition, integrated into the attention-based network architecture, is particularly well-suited for the large-scale point cloud segmentation.

### 3.2.2 Surface feature extractor

For the target point  $p(x_p, y_p, z_p)$ , the local surface features are extracted by defining a fixed-radius neighborhood  $B(p, r)$ , where  $r$  is the radius of the ball query. The covariance matrix of the region  $B(p, r)$  is constructed as follows:

$$M_{3 \times 3} = \frac{1}{k} \sum_{i=1}^k (p_i - p)(p_i - p)^T \quad (3.11)$$

where  $p_i$  denotes the neighborhood point of the point  $p$ , and  $k$  is the total number of points in the adjacent area  $B(p, r)$ . The covariance matrix  $M_{3 \times 3}$  characterizes the local geometric distribution of the point  $p$  in the neighborhood. Subsequently, by performing eigen decomposition on the matrix  $M_{3 \times 3}$ , the principal components including the eigenvectors and their corresponding eigenvalues can be calculated as follows:

$$M_{3 \times 3} = E \Lambda E^T \quad (3.12)$$

where each column of the matrix  $E$  is an eigenvector of  $M_{3 \times 3}$ , and the diagonal elements of the matrix  $\Lambda$  are the corresponding eigenvalues, which are denoted as  $(e_1^M, e_2^M, e_3^M)$  and  $(\lambda_1^M, \lambda_2^M, \lambda_3^M)$ , respectively. Note that the eigenvalues are sorted in descending order, namely  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ . Eigenvectors represent the principal directions of distribution within the point cloud  $B(p, r)$ , while eigenvalues quantify the degree of variation along the corresponding directions. In the context of the least-squares fitting, the local region is approximated by a planar structure, where the normal vector of the plane is defined as the eigenvector corresponding to the smallest eigenvalue.

Therefore, the derived principal components can be employed to further construct a set of features that characterize the intrinsic properties of the surface structure. The surface features of the point  $p$  are then calculated as follows:

$$\begin{cases} n_p = e_3^M \\ d_p = -n_p \cdot p \\ s_p = \frac{\lambda_3^M}{\lambda_1^M} \\ c_p = \frac{\lambda_3^M}{\lambda_1^M + \lambda_2^M + \lambda_3^M} \\ r_p = \frac{1}{k} \sum_{i=1}^k \frac{|n_p \cdot p_i + d_p|}{\|n_p\|} \end{cases} \quad (3.13)$$

where  $n_p$  represents the normal vector  $(n_{px}, n_{py}, n_{pz})$  and  $d_p$  denotes the distance from the origin to the fitted plane. It should be noted that the surface roughness  $r_p$  is quantified as the mean residual of all points in the neighborhood relative to the fitted plane. The extracted surface features composed of low-level descriptors encapsulate rich local information that serves as the foundation for subsequent processes.

### 3.2.3 Local context embedding

The surface feature extractor, designed as an independent module, generates point-wise geometric context features with high flexibility. It can be seamlessly integrated as a loosely coupled component into the attention network architecture introduced in Section 3.1, enhancing the network's feature representation capabilities. The embedding process is illustrated in Fig. 3.5. The inclusion of the surface contextual embedding (SCE) enhance the framework's ability to capture subtle variations and intricate interactions across the surface. Moreover, the adaptive nature of the self-attention mechanism ensures that the integration process dynamically adjusts to different surface structures, making it particularly effective for fine-grained segmentation tasks.

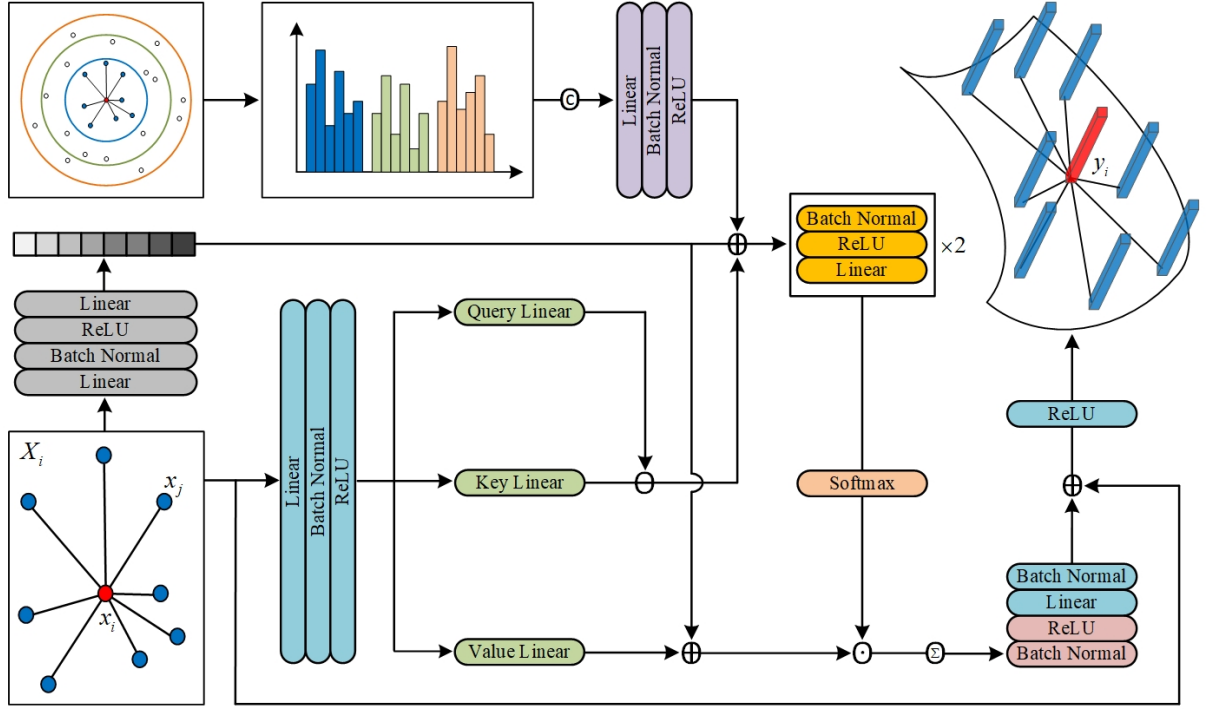


Figure 3.5: Illustration of the local surface context embedding. By incorporating neighborhoods of multiple scales, the embedding preserves multi-scale information from the original surface.

To address the potential loss of local geometric information caused by sampling operations in the hierarchical structure, the surface feature extraction module is designed to operate before sampling. For large-scale point clouds, multi-scale surface features can be extracted as an extended variant, ensuring that the backbone network is provided with comprehensive and original geometric context information. The extracted surface features are then directly concatenated with the original point positions along the channel dimension and fused through an MLP before being fed into the backbone network. The whole embedding process is represented by the following formula:

$$f_p^n = \gamma \left( f_p^{r_1} \textcircled{\text{C}} f_p^{r_2} \textcircled{\text{C}} \dots \textcircled{\text{C}} f_p^{r_n} \right) \quad (3.14)$$

where  $f_p^n$  represents the fused feature of the point  $p$  encapsulating multi-scale local context. The term  $f_p^{r_i}$  refers to the surface feature extracted at the radius  $r_i$ , and the operator  $\textcircled{\text{C}}$  signifies the feature concatenation along the channel dimension. The function  $\gamma(\cdot)$  is an MLP consisting of a linear layer, a batch normalization layer, and a ReLU activation layer, which is responsible for the compatibility with the input size of the subsequent network. Within the entire neighboring region, the multi-scale spatial geometric relationships of point  $p$  are explicitly embedded into  $f_p^n$ ,

which allows the resulting point features to consistently preserve structural integrity and semantic richness. The whole embedding process ensures effective collaboration between fine-grained local details and global features in the hierarchical network.

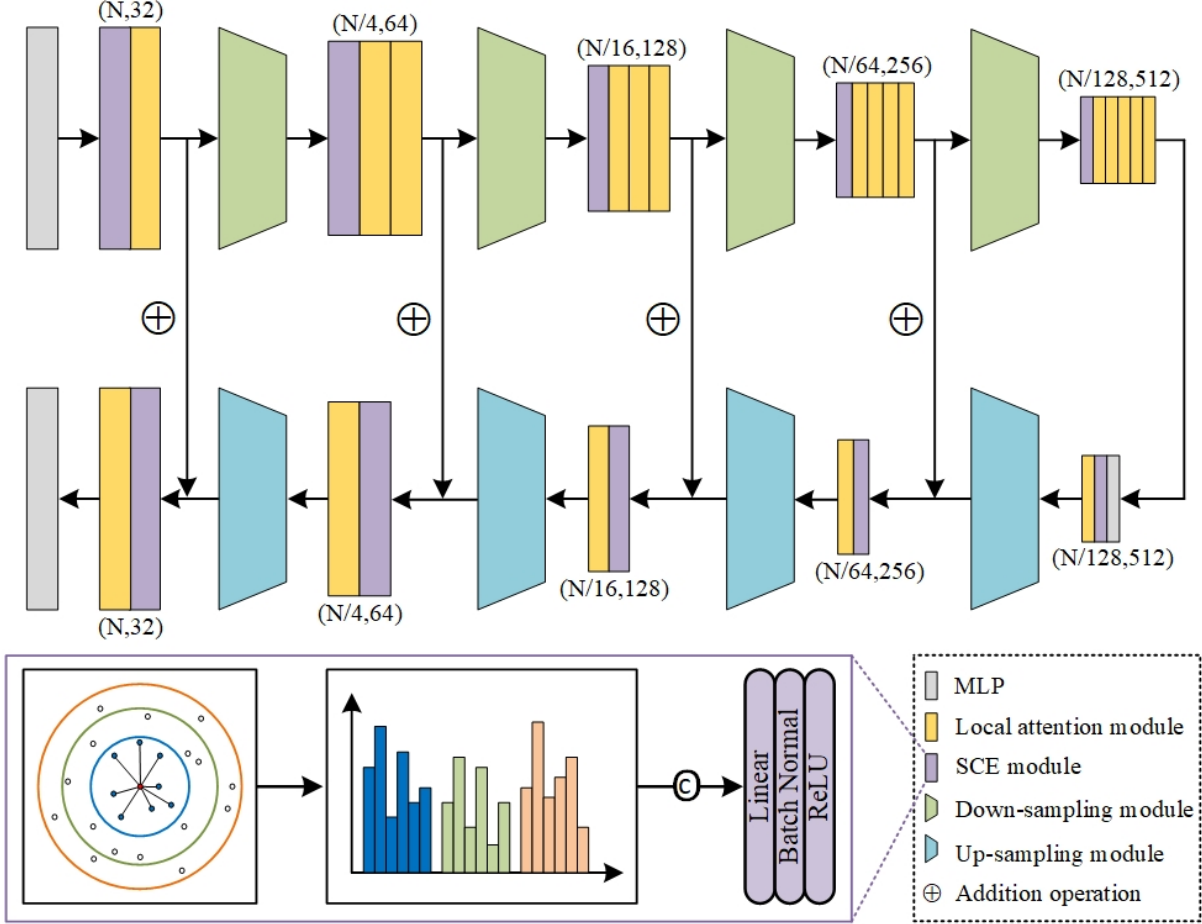


Figure 3.6: Detailed architecture of the local context embedding network. The SCE module is loosely coupled with the attention aggregation framework, which generates multi-scale point-wise geometric context features with high flexibility.

Building upon the method presented in Section 3.1.1, we deeply embed surface contextual features into the local attention aggregation framework to enhance the representational capacity of the self-attention mechanism. For each input point  $x_i$  and its local region  $X_i$ , the attention score that includes surface features  $f_p^n$  is calculated as follows:

$$a_i = \sum_{x_j \in X_i} \alpha \left( \beta (q_i - k_j + p_{i,j} + f_i^n) \right) \odot (v_j + p_{i,j}) \quad (3.15)$$

where  $q_i$ ,  $k_j$  and  $v_j$  represent the query, key and value embeddings of the input point  $x_i$  and its neighboring point  $x_j$ , respectively. The term  $p_{i,j}$  denotes the positional encoding between  $x_i$  and  $x_j$ , which provides relative position information for the attention mechanism. The function  $\beta(\cdot)$  is a learnable transformation that generates correlation vectors for feature aggregation, while  $\alpha(\cdot)$  normalizes attention weights using a softmax operation. Their specific implementations are consistent with the description in Section 3.1. Similarly, a residual module is also incorporated to produce the final output attention feature as described in Equation 3.6.

Unlike classic fusion methods that treat local surface features solely as point-wise accessory channels, our approach integrates them more comprehensively into the attention framework. The details of the network structure are shown in Fig. 3.6. On one hand, by incorporating the attention mechanism to dynamically aggregate local features, the progressive stacking of sampling modules significantly expands the network’s receptive field. This allows the network to cover unsampled points and generate rich contextual features for them. On the other hand, for local details with complex structures that are blurred by down-sampling operations, the embedding of the original surface features effectively compensates for the information loss introduced during the sampling process. The network parameters are updated by optimizing the cross-entropy loss function.

The introduction of surface features provides the fused features with initial fine-grained contextual support, enabling the attention network to better capture subtle local geometric variations when constructing attention weights, particularly in regions with intricate details and complex boundaries. The loosely coupled design of the surface feature extractor, combined with its efficient feature embedding process, offers a flexible and universal solution for fine-grained segmentation tasks in large-scale point clouds.

### 3.3 Contextual joint augmentation for rare classes

In large-scale point cloud segmentation tasks, the segmentation of rare categories presents substantial challenges due to inherent data imbalance and the complexity of real-world scenes. These rare categories are often characterized by sparse distributions and insufficient diversity in their forms of representation, which limits the model’s ability to generalize effectively to unseen scenarios. This issue is further exacerbated by the intricate contextual environments in which these rare objects are typically embedded, making it difficult for models to distinguish and segment them accurately. Addressing this challenge requires a strategy that not only enhances the representation of rare objects but also incorporates their contextual surroundings to provide a holistic understanding.

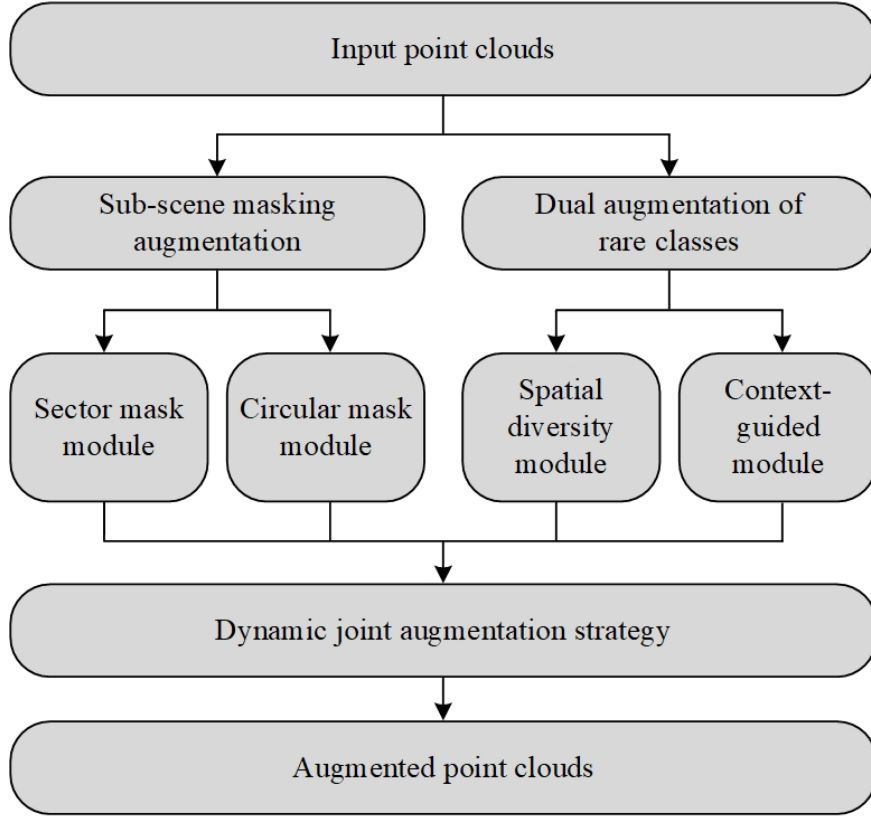


Figure 3.7: Workflow of contextual joint augmentation. The framework combines two complementary augmentation branches at the scene and class levels, where the modular components are integrated through a dynamic joint strategy to generate structurally consistent and semantically diverse augmented point clouds.

To this end, we introduce a context-guided joint augmentation strategy, which systematically diversifies the training point cloud data from two dimensions: the rare objects themselves and their surrounding environments. In Fig. 3.7, we illustrate the core augmentation modules we designed, along with the workflow of their integration. In detail, the method mainly includes sub-scene mask augmentation and context-guided augmentation of rare targets. A modular design combined with a dynamic mixing mechanism is then utilized to expand the spatial distribution of rare objects and enhance their interaction with the contextual features of the scene. By integrating these augmented features into the training pipeline, the method improves the model’s ability to perceive and adapt to complex contextual relationships, particularly in scenarios where rare objects and intricate scene structures coexist. By addressing both the diversity of rare class representations and their contextual cues, this augmentation strategy significantly enhances segmentation performance and robustness for rare class in large-scale point clouds with complex and highly variable environments.

### 3.3.1 Sub-scene masking augmentation

Rare categories are often deeply embedded within complex point cloud environments, making it critical to enhance the surrounding context to effectively capture their unique contextual features. To address this, we propose a sub-scene masking augmentation strategy that begins by partitioning large-scale point clouds into overlapping sub-scenes. These sub-scenes are then combined using masking templates to generate diverse contextual environments. To further simulate the variety of environmental conditions encountered in real-world scenarios, we introduce two specialized sub-modules: the sector mask module and the circular mask module. These modules are designed to expand the model’s capacity for perceiving and adapting to different contextual patterns, thereby improving its robustness.

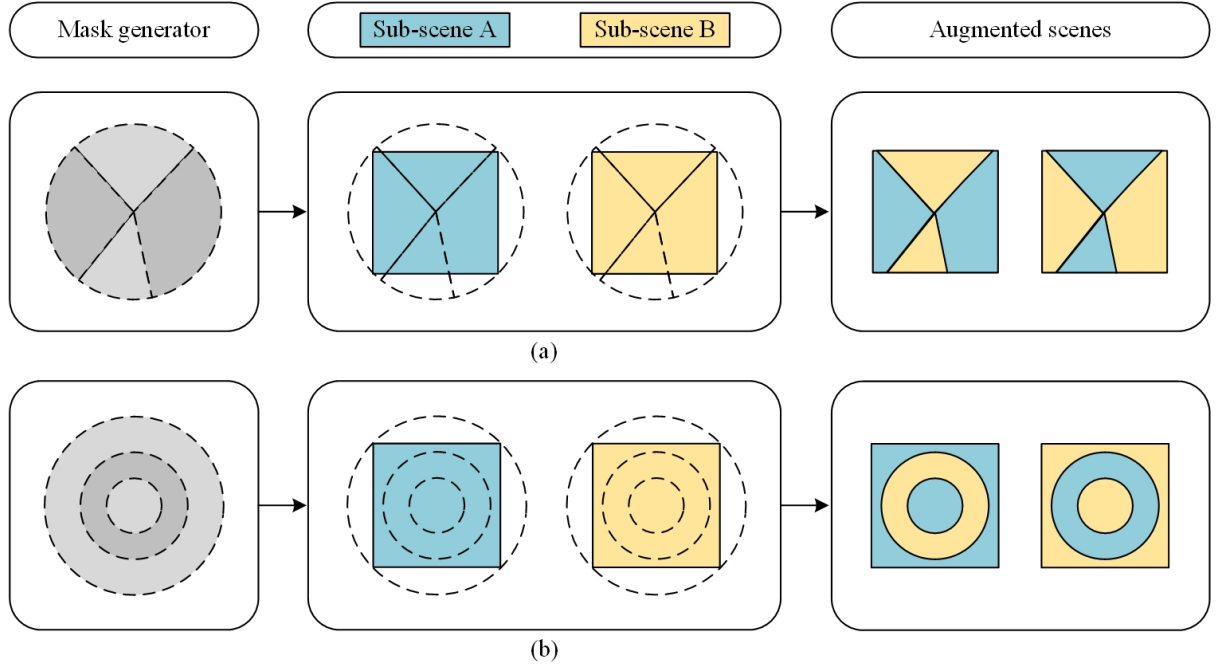


Figure 3.8: Illustration of the sub-scene masking augmentation. (a) shows the workflow of the sector mask module; (b) presents the workflow of the circular mask module.

#### Sector mask module

The sector mask module enhances the scene environment by segmenting sub-scenes based on yaw angles and simulating real-world conditions such as incomplete views or constrained perspectives through random occlusion or replacement of sector regions, as shown in Fig. 3.8(a). The process begins by taking the center of the sub-scene as the origin and generating divergent rays based

on random yaw angles. These rays divide the sub-scene into distinct sector-shaped regions, which are then used to create masking templates. The generated sector masks are applied to sub-scenes containing similar categories, dividing them into complementary sector regions. Subsequently, alternate sectors from different sub-scenes are selected and fused to introduce new environmental distributions, resulting in the creation of novel sub-scenes. These newly generated, diverse contextual environments enhance the model’s ability to generalize different categories under varying environmental conditions. By simulating occlusion and varying viewpoints, the Sector Mask Module improves the model’s adaptability to incomplete or dynamically changing contexts.

### Circular mask module

The circular mask module focuses on multi-level integration of the surrounding environment with distinct structural elements, simulating the layered fusion of foreground and background commonly observed in real-world scenarios, as presented in Fig. 3.8(b). This module generates circular masking templates by dividing the sub-scene into concentric ring-shaped regions based on predefined radial distances. Each ring represents a different layer of foreground and background, allowing for precise control over the contextual composition. These circular masks are applied to sub-scenes containing rare categories, enabling the isolation of specific foreground and background regions. Points within the ring regions are alternately replaced with points from other sub-scenes to dynamically adjust the local context distribution around the target points. This multi-level adjustment can refine the local contextual information surrounding different categories, and further focus on the interaction of local foreground and background in complex sub-scene regions.

Together, the sector mask module and circular mask module provide complementary approaches to augmenting the scene context. The sector mask module diversifies the distribution of large-scale scene environments by introducing novel angular divisions, while the circular mask module fine-tunes the layered relationship between foreground and background. By integrating these modules, the proposed strategy can effectively enhance the contextual diversity of different categories, significantly improving the model’s ability to generalize and adapt to complex real-world scenarios.

### 3.3.2 Dual augmentation of rare classes

Enhancing rare categories aims to address the challenges posed by data imbalance while improving the representation of rare targets under varying semantic and spatial conditions. By increasing the frequency and diversity of rare class appearances, the model can better learn robust feature representations for these categories. This paper introduces a novel dual augmentation approach that combines a spatial diversity augmentation module and a context-guided selection module, enhanc-

ing the representation of rare categories from two dimensions: spatial distribution and semantic consistency.

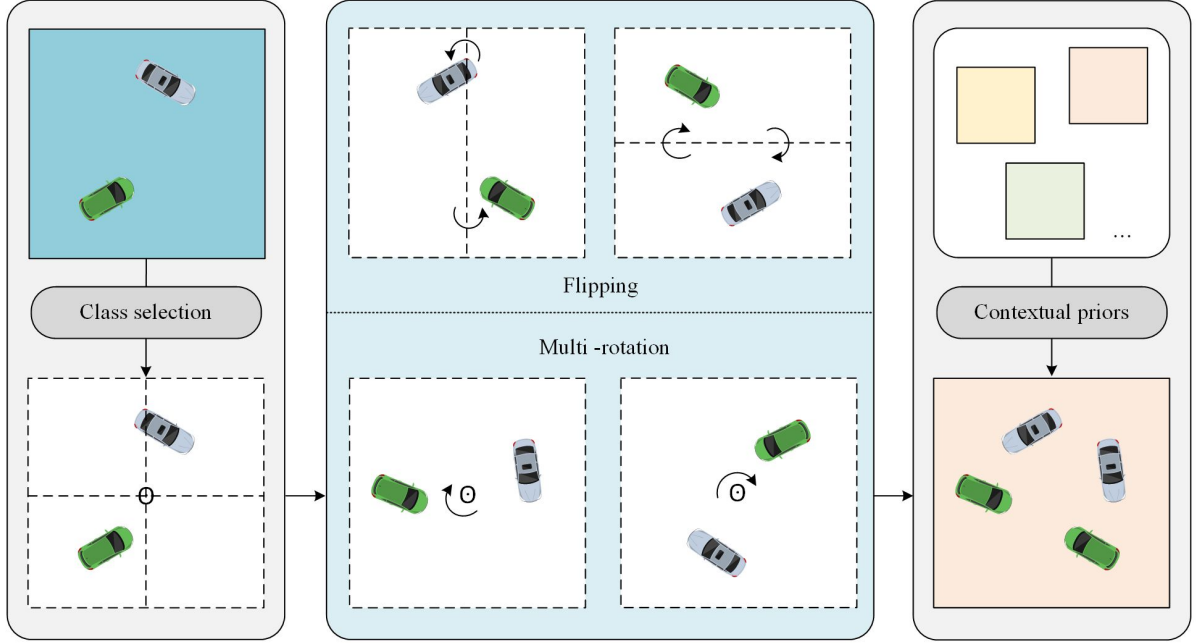


Figure 3.9: Illustration of the dual augmentation of rare classes. Rare class instances are augmented via flipping and multi-rotation operations, then reintegrated into scenes using contextual priors to ensure semantic consistency.

### Spatial diversity augmentation module

The spatial diversity augmentation module focuses on expanding the spatial distribution of rare categories. As illustrated in Fig. 3.9, it begins by extracting the target point cloud of rare categories from sub-scenes containing such objects. The extracted point cloud is then subjected to multiple rotation transformations around the Z-axis, generating representations of rare classes from various angles. In addition to rotation, the module incorporates flipping operations along the X-axis or Y-axis to further increase variability. This mirroring transformation simulates scenarios where objects are observed in mirrored spatial configurations, such as reflections in real-world environments or unconventional viewpoints. After applying these transformations, the augmented objects are pasted into different scenes at random locations. The whole process not only expands the number of samples for rare categories, but also simulates their potential appearances with varied orientations and symmetries. By introducing greater variability in spatial configurations and perspectives,

the module can enhance the model’s ability to generalize rare category representations to unseen scenarios.

### Context-guided selection module

The context-guided module leverages prior knowledge of surrounding environments to ensure the semantic coherence between rare classes and their associated context. This module focuses on constructing realistic and semantically consistent combinations of rare objects and their contextual environments, as shown in Fig. 3.9. To achieve this, we predefine several strongly correlated contextual categories for each rare class. Sub-scenes are then filtered based on these constraints, ensuring that augmentation operations are applied only when the corresponding contextual categories are detected in the scene. For example, for a chimney, its predefined contextual categories might include rooftops and building facades. Augmentation for chimneys is applied only when contextual information associated with buildings is present in the scene. This strategy ensures that rare categories are placed in realistic environments that mimic real-world semantic relationships, enabling the model to better understand and learn the interplay between rare objects and their context.

By combining these two sub-modules, the proposed approach achieves a dual enhancement: it extends the spatial distribution of rare categories while preserving the semantic relationships that characterize their interactions with their environments. The spatial diversity augmentation module increases sample diversity, while the context-guided module ensures the augmented samples align with realistic contextual relationships. Together, these modules effectively simulate real-world conditions, improving the recognition performance of rare categories in large-scale point cloud segmentation tasks.

#### 3.3.3 Dynamic joint augmentation strategy

To fully integrate the capabilities of the aforementioned modules, this paper proposes a dynamic joint augmentation strategy designed to construct diverse scenarios for rare categories based on contextual information. The core idea of this joint strategy is to achieve comprehensive augmentation for rare categories through the random combination of sub-modules from both sub-scene augmentation and rare class augmentation. Specifically, the joint augmentation process begins by using a randomized combination of the sector mask module and the circular mask module to enrich the diversity of surrounding environments containing rare categories. Subsequently, it applies the spatial diversity augmentation module and the context-guided selection module to expand the spatial and semantic distributions of rare categories in a targeted manner. The multi-level integration of both sub-scene and rare class augmentation ensures that rare categories are not only well-represented

across diverse spatial configurations but also embedded within intricate and challenging contextual information for the surrounding environments.

Futhermore, this joint strategy incorporates a dynamic augmentation mechanism, which allows the relative contributions of each module to the overall augmentation process to be balanced through configurable weights. This enables efficient customization of the augmentation pipeline to suit specific application scenarios. For example, in tasks that rely heavily on contextual understanding of the scene, a greater emphasis can be placed on the scene-level augmentation modules. Conversely, for tasks that require a greater focus on enhancing the diversity of rare categories, the class-level augmentation modules can be prioritized. By balancing the sub-scene augmentation and rare class augmentation, this dynamic mixing operation can ensure that the augmented sub-scenes are generated from a wide range of spatial distributions and semantic contexts, offering robust support for segmenting rare categories under varying conditions.

The proposed contextual joint augmentation strategy provides a comprehensive framework for enhancing the representation of rare categories in large-scale point cloud segmentation tasks. By systematically diversifying the spatial distribution of rare classes and their surrounding environments, the strategy significantly improves the the generalization ability beyond the contextual priors of the training scenes and adapt to complex real-world scenarios. The modular design and dynamic mixing mechanism offer a flexible and scalable solution to the challenges of rare class segmentation in large-scale point clouds.

### 3.4 Target-aware attention for rare class segmentation

The distinctive shape characteristics inherent in target objects play a critical role in semantic segmentation. These features not only enable the precise identification of contour attributes specific to each rare target on an individual level but also contribute to improving the structural representation of rare classes (Zhang et al., 2023c). This instance-level focus is essential for addressing the challenges posed by the sparse distribution and unique geometry of rare objects in point cloud data. To this end, we propose a target-aware aggregator that leverages the spatial distribution of rare classes to enhance the segmentation of unique targets. By encapsulating these spatial characteristics at the instance level, semantic segmentation models can achieve a more nuanced and detailed understanding of rare classes, enhancing their ability to distinguish these unique targets.

#### 3.4.1 Pseudo labels for rare targets

To represent the spatial and geometric characteristics of a rare target, we introduce a descriptor in the form of a 7-tuple  $\{w_i, h_i, l_i, u_i, v_i, d_i, r_i\}$ . This descriptor captures both the shape and positional

attributes of the object, providing a compact yet informative representation for segmentation tasks. The tuple  $\{w_i, l_i, h_i\}$  encodes the simplified physical dimensions of a rare target, effectively abstracting its overall geometric shape. This abstraction balances detail and computational efficiency, offering a descriptive yet manageable outline of the object’s essential features without unnecessarily complicating the model. Given the complexity of the 3D target representation, the positional attributes  $\{u_i, v_i, d_i, r_i\}$  are equally vital. These elements capture detailed spatial information about the target’s ground geometric center and the heading of a target. Specifically,  $\{u_i, v_i, d_i\}$  define the precise spatial location of the target, while  $\{r_i\}$  articulates the target’s orientation and alignment in the overall scene. This combination allows the segmentation framework to achieve a more holistic understanding of rare targets by integrating their size, location, and orientation into the learning process.

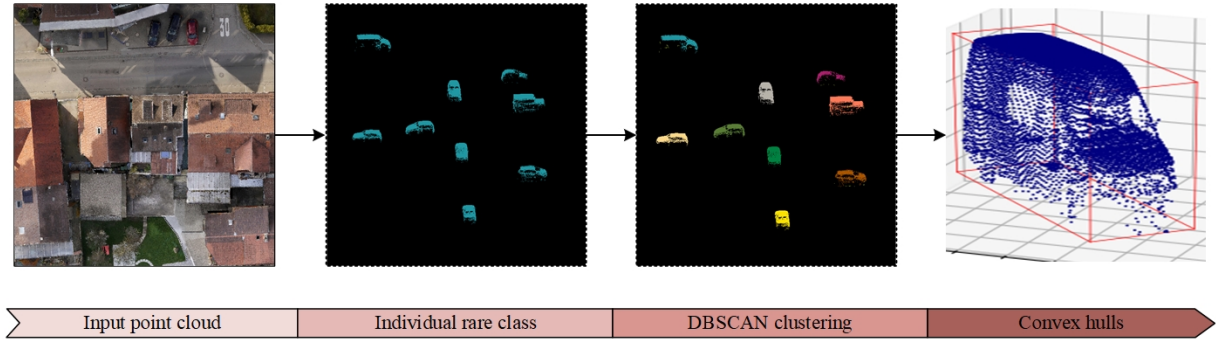


Figure 3.10: Workflow of the pseudo label generation for rare targets.

To provide supervisory signals for the training of the proposed target-aware aggregator, we employ a multi-step process to generate pseudo-labels tailored to rare classes, ensuring clarity and minimizing confusion with majority classes, as illustrated in Fig. 3.10. First, rare classes are individually isolated based on semantic labels to avoid the confusion with common categories. Considering the discrete and sparse distribution of rare targets at the instance level, the density-based spatial clustering of applications with noise (DBSCAN) method is applied to separate the point cloud of rare objects into distinct, reliable clusters. This clustering step is crucial for identifying coherent groups of points that belong to specific rare objects. Subsequently, the convex hulls of the identified clusters are calculated, serving as an approximation of the target’s structural boundaries. These convex hulls are then adjusted and aligned with the vertices of the target’s structure to serve as supervision signals for training the learnable structural target representation.

### 3.4.2 Target feature encoding module

In order to design a lightweight aggregator while ensuring computational efficiency, the target feature encoding (TFE) module is introduced, operating on 3D points with local attention features. The TFE module effectively aggregates local features and embeds them into a compact representation for downstream processing.

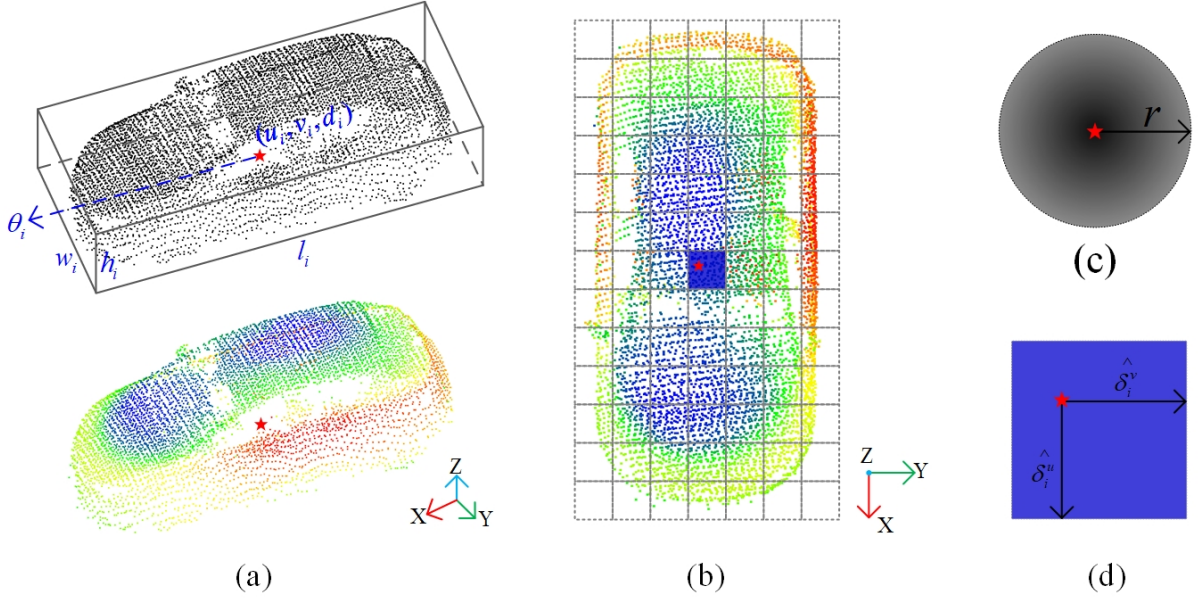


Figure 3.11: Illustration of the proposed target-aware aggregator. Here, the red star indicates the center of the target instance. (a) shows the structural characteristics of a rare target and the corresponding attention features; (b) presents the target feature encoding module from top view; (c) illustrates the Gaussian kernel strategy to boost the supervisory signal; (d) depicts the compensation for quantization error in the target topology parsing module.

Specifically, the local attention features of points in Fig. 3.11(a) are partitioned into different voxels based according to their spatial coordinates. Within each voxel, the features of all points are stacked into a feature set, which is subsequently averaged to form a representative voxel-level feature. This averaging process reduces the computational burden while preserving the essential features of the points within each voxel. The resulting voxel feature is then embedded into a vector of a given dimension using an MLP layer, ensuring compatibility with the input size of subsequent processes. The voxel feature encoding can be formulated as:

$$f^{i,j,k} = \chi(m(a_1^{i,j,k} \cup a_2^{i,j,k} \cup \dots \cup a_t^{i,j,k})) \quad (3.16)$$

where  $f^{i,j,k}$  is the encoded feature belonging to the voxel indexed by  $\{i, j, k\}$ , and  $t$  represents the total number of points in this voxel. The term  $a_t^{i,j,k}$  refers to the local attention feature of the  $t_{th}$  point in the voxel. The symbol  $\cup$  denotes the stacking operation of elements, while the function  $m(\cdot)$  represents the mean pooling used to aggregate features within the voxel. The embedding layer, denoted by  $\chi$ , consists of a linear layer, a batch normalization layer, and a ReLU activation layer. Following the voxel-level encoding, the features along the height dimension are flattened to generate a top-view feature tensor  $X \in \mathbb{R}^{L \times W \times C}$ , as illustrated in Fig. 3.11(b). Such flattening process aims to eliminate height redundancy, significantly improving computing efficiency by reducing the dimensionality of feature representation. Additionally, this top-view representation narrows the search space for rare targets, focusing on the critical planar spatial distribution of the points.

### 3.4.3 Candidate-aware mask module

In order to accurately determine the center position of rare categories, the feature map is passed forward to the candidate-aware mask (CAM) module to perform the candidate selection task of center points, as depicted in Fig. 3.12.

For  $K$  different categories of rare objects, the feature map  $X$  is duplicated to separate center point identification for each category individually. This allows the network to handle center point selection across multiple rare classes with distinct features. The operation of the CAM module can be represented as:

$$Y^{L \times W \times K} = g_1(X^{L \times W \times C}) \odot g_2(X^{L \times W \times C}) \odot \dots \odot g_K(X^{L \times W \times C}) \quad (3.17)$$

where  $g_k(\cdot)$  is a shared MLP tailored for the  $k_{th}$  rare class, which is composed of a normalization layer and a logistic activation layer. The results of different classes are then concatenated, represented by the operation  $\odot$ . The final output  $Y^{L \times W \times K}$  is a multiple channel heatmap, where  $K$  denotes the number of center point types. Each channel corresponds to a distinct rare category, and its values represent the likelihood of points being the center of the respective class.

Rare categories in large-scale scenes are typically characterized by their infrequency, leading to the sparse distribution of center points on the feature maps. To address this issue, we propose an enhancement strategy based on the Gaussian kernel to generate supervisory signals, which are generated by projecting the 3D centers of rare targets onto a top-view map, as shown in Fig. 3.11(c). For a given center point belonging to the  $k_{th}$  class, it is treated as the initial reference anchor. A Gaussian function is then used to render it into a continuous kernel, confined within a predefined radius  $r$ , as follows:

$$K_r = e^{-\frac{\{x-c_x\}^2 + \{y-c_y\}^2}{2\sigma_k^2}} \quad (3.18)$$

where  $\sigma_k$  is the adaptive standard deviation which is determined by the size of the  $k_{th}$  target object, and  $c_x$  and  $c_y$  represents the target center.

In the training phase, the modified focal loss combined with the center-anchored Gaussian kernel is chosen to update the network as follows:

$$loss_{mask} = -\frac{1}{N} \sum_{i=1}^N \begin{cases} (1 - \hat{p}_i)^\gamma \log(\hat{p}_i) & \text{if center} \\ (1 - K_r)^\alpha (\hat{p}_i)^\gamma \log(1 - \hat{p}_i) & \text{otherwise,} \end{cases} \quad (3.19)$$

where  $\hat{p}_i$  represents the predicted confidence score of a candidate center point, which can be also employed to adapt and refine the local attention scores dynamically. The parameters  $\alpha$  and  $\gamma$  are modulating factors of the loss function, which control the contribution of each point. We set  $\gamma = 2$  and  $\alpha = 4$  in all our experiments. The term  $(1 - K_r)^\alpha$  reduces the penalty for meaningful points strongly correlated with the actual centers, thereby allowing the model to focus on regions of interest without over-penalizing surrounding areas. This approach effectively spreads the influence of each center point to nearby areas, generating a dense supervisory signal that compensates for the inherent sparsity of rare class center points.

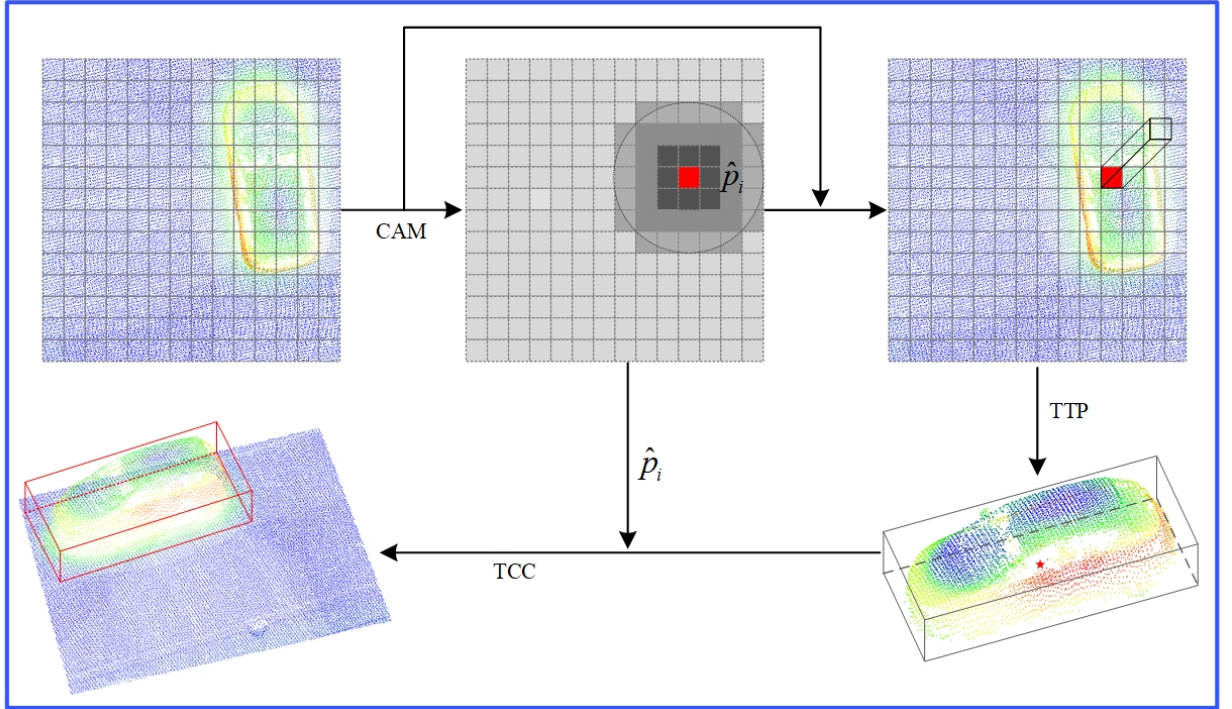


Figure 3.12: Workflow of the proposed target-aware aggregator. The blue box represents the aggregator for a single category. CAM: Candidate-aware mask module. TTP: Target topology parsing module. TCC: Target confidence constraint module.

### 3.4.4 Target topology parsing module

To fully leverage the spatial discrepancies between independent instances, we design the target topology parsing (TTP) module, which explicitly explain the structural characteristics of targets, as depicted in Fig. 3.12. The module highlighted with the blue box in this figure is also depicted in conjunction with Fig. 3.14. For each positive feature located at a center point, the TTP module regresses a seven-dimensional vector  $\hat{t}_i$  that represents the structural characteristics of a rare target at the instance level. This vector can be formulated as:

$$\hat{t}_i = \{\hat{w}_i, \hat{l}_i, \hat{h}_i, \hat{\delta}_i^u, \hat{\delta}_i^v, \hat{d}_i, \sin(\hat{\theta}_i)\} \quad (3.20)$$

where  $\{\hat{w}_i, \hat{l}_i, \hat{h}_i\}$  represents the 3D size of the target, including its width, length, and height. The value  $\hat{d}_i$  provides the target's vertical center location along the  $Z$  axis. During the voxelization process, intricate geometric details are often lost due to the discretization of points. To compensate for these quantization errors, we include offset terms  $\{\hat{\delta}_i^u, \hat{\delta}_i^v\}$ , which act as corrections for the positional inaccuracies caused by discretization of points, as shown in Fig. 3.11(d). In the context of the top-view feature map, the orientation of the target undergoes a further simplification into the yaw angle  $\theta_i$ , representing the object's heading on the ground plane. This reduction not only provides a straightforward yet effective way to denote the target's directional facing, but also aligns with the human intuitive understanding of spatial orientation, simplifying the complexity of 3D orientation data. Moreover, distinguishing between positive and negative orientations is often redundant for 3D segmentation task. Therefore only the sine of the yaw angle  $\sin(\hat{\theta}_i)$  is retained to encode the orientation, which suffices to capture the necessary directional information.

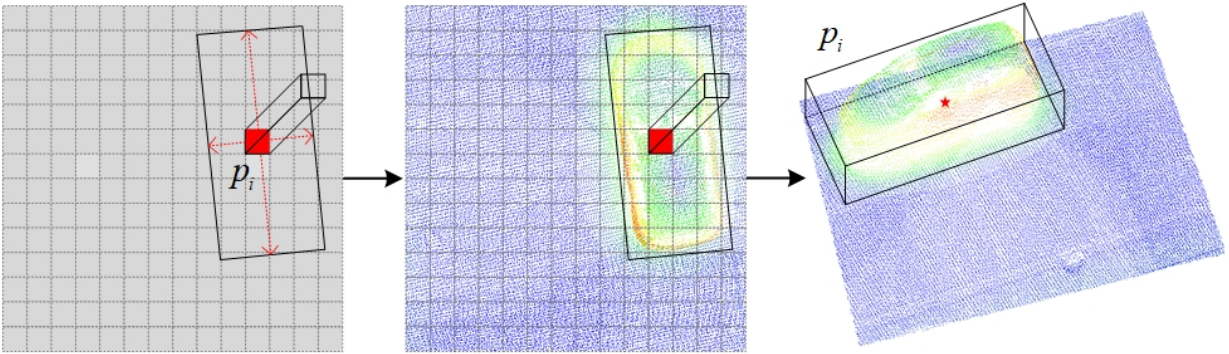


Figure 3.13: Illustration of the target confidence constraint module.

Each regression task is assigned its own specific output head, ensuring tailored learning of

structural parameters for different targets. All parameters in the structural representation vector  $\hat{t}_i$  are trained using the ground truth centers and supervised with the L1 Loss as follows:

$$\begin{aligned}
loss_{ins} = \frac{1}{N} & \left( \sum_{i=1}^N \sum_{\hat{s} \in (\hat{w}_i, \hat{h}_i, \hat{h}_i)} |\hat{s}_i - s_i| \right. \\
& + \alpha \sum_{i=1}^N \sum_{\hat{o} \in (\hat{\delta}_i^u, \hat{\delta}_i^v)} |\hat{o}_i - o_i| \\
& + \beta \sum_{i=1}^N |\hat{d}_i - d_i| \\
& \left. + \gamma \sum_{i=1}^N |\sin(\hat{\theta}_i) - \sin(\theta_i)| \right)
\end{aligned} \tag{3.21}$$

where  $N$  is the total number of target objects, the parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  are the weight factors that fine-tune the balance between the regression losses of size, quantization error,  $Z$  location and yaw orientation.

### 3.4.5 Target confidence constraint module

In order to obtain reliable confidence scores associated with target centers during the testing phase, we additionally design the target confidence constraint (TCC) module, which aims to identify and refine peaks on the heatmap, as illustrated in Fig. 3.13. This module enhances the precision of center point selection, ensuring that subsequent target parsing focuses on meaningful locations.

As described in Section 3.4.3, the output heatmap  $Y^{L \times W \times K}$  can be interpreted as  $K$  binary feature masks with the same size of  $L \times W$ , and each of them is uniquely corresponding to the positional information of one specific class. For each category, we initially identify the top- $K$  confidence scores as candidate peaks on the heatmap, which indicate the potential object centers associated with the corresponding class. To further refine the quality of the selected peaks, we apply a pre-defined threshold on the confidence scores, retaining only those peaks that meet the criteria. This step eliminates low-confidence candidates, ensuring that only high-quality peaks are considered as target centers. Each peak in a resulting dense heatmap corresponds to a specific target center.

The refined heatmap after the filtering operation is denoted as  $\hat{Y} \in [0, 1]^{L \times W \times K}$ , representing a binary mask where positive values indicate identified center points. Each predicted center point in  $\hat{Y}$  is associated with exactly one positive ground-truth location, while all other locations are treated as negative. Thus, we utilize the binary mask  $\hat{Y} \in [0, 1]^{L \times W \times K}$  to filter the feature map  $X \in \mathbb{R}^{L \times W \times C}$  described in Section 3.4.2. This ensures that only the features corresponding to

the heatmap peaks are retained, while irrelevant or low-confidence regions are discarded. In the testing phase, the parsing process of targets operates exclusively on features associated with positive center points. Since distinct entities in real-world scenes cannot occupy the same physical space, non-maximum suppression (NMS) is employed to remove redundant targets with high confidence scores, as described in Algorithm 2. The operation ensures that only the most relevant and non-overlapping targets are preserved, improving the reliability of the predicted centers.

---

**Algorithm 2** Non-maximum suppression for redundant bounding boxes of targets

---

**Input**

- A set of bounding boxes  $B = \{b_i\}_{i=1}^N$ , where each box  $b_i = (x_i, y_i, z_i, l_i, w_i, h_i, \theta_i)$
- Confidence scores  $\{s_i\}_{i=1}^N$  for each bounding box
- Intersection over Union (IoU) threshold  $T$

```

1: Initialize the output set of selected bounding boxes:  $S \leftarrow \emptyset$ 
2: Project each 3D bounding box  $b_i$  to the top view plane to obtain 2D bounding boxes:  $\hat{B} = \{\hat{b}_i\}_{i=1}^N$ , where  $\hat{b}_i = (x_i, y_i, l_i, w_i, \theta_i)$ 
3: Sort  $\hat{B}$  by confidence scores  $\{s_i\}$  in descending order
4: while  $\hat{B} \neq \emptyset$  do
5:   Select the bounding box  $\hat{b}^*$  with the highest confidence score from  $\hat{B}$ :  $\hat{b}^* = \arg \max_{\hat{b}_i \in \hat{B}} (s_i)$ 
6:   Add the corresponding 3D bounding box  $b^*$  to  $S$ :  $S \leftarrow S \cup \{b^*\}$ 
7:   Remove  $\hat{b}^*$  from  $\hat{B}$ :  $\hat{B} \leftarrow \hat{B} \setminus \{\hat{b}^*\}$ 
8:   for each remaining  $\hat{b}_j$  in  $\hat{B}$  do
9:     Calculate the IoU between  $\hat{b}^*$  and  $\hat{b}_j$ :  $\text{IoU}(\hat{b}^*, \hat{b}_j)$ 
10:    if  $\text{IoU}(\hat{b}^*, \hat{b}_j) \geq T$  then
11:      Remove  $\hat{b}_j$  from  $\hat{B}$ :  $\hat{B} \leftarrow \hat{B} \setminus \{\hat{b}_j\}$ 
12:    end if
13:  end for
14: end while
15: return  $S$ 

```

**Output**

- The set of selected bounding boxes  $S$  after NMS
- 

### 3.4.6 Target focus aggregation module

To make full use of the benefits of local focus attention and instance-level topology, we further introduce the target focus aggregation (TFA) module to integrate features across different domains. This module enhances the model's ability to capture rare target characteristics while maintaining a balanced focus between common and rare classes.

Building upon Eq. 3.20, which encapsulates the spatial attributes of rare targets, the TFA module remaps the corresponding regions on the local attention scores  $A \in \mathbb{R}^{N \times C}$  by combining these spatial attributes with the coordinates of point-wise attentions. Following this, the confidence scores filtered by the TCC module  $\hat{P} \in \mathbb{R}^{L \times W \times K}$  serves as weighting factors to modulate the network's attention. By combining the spatial remapping and the confidence-based weighting, the TFA module ensures that the attention mechanism incorporates both the local geometric relationships and the instance-level topology of rare classes, thereby enhancing the feature representation of rare classes under the infrequency and sparse distribution.

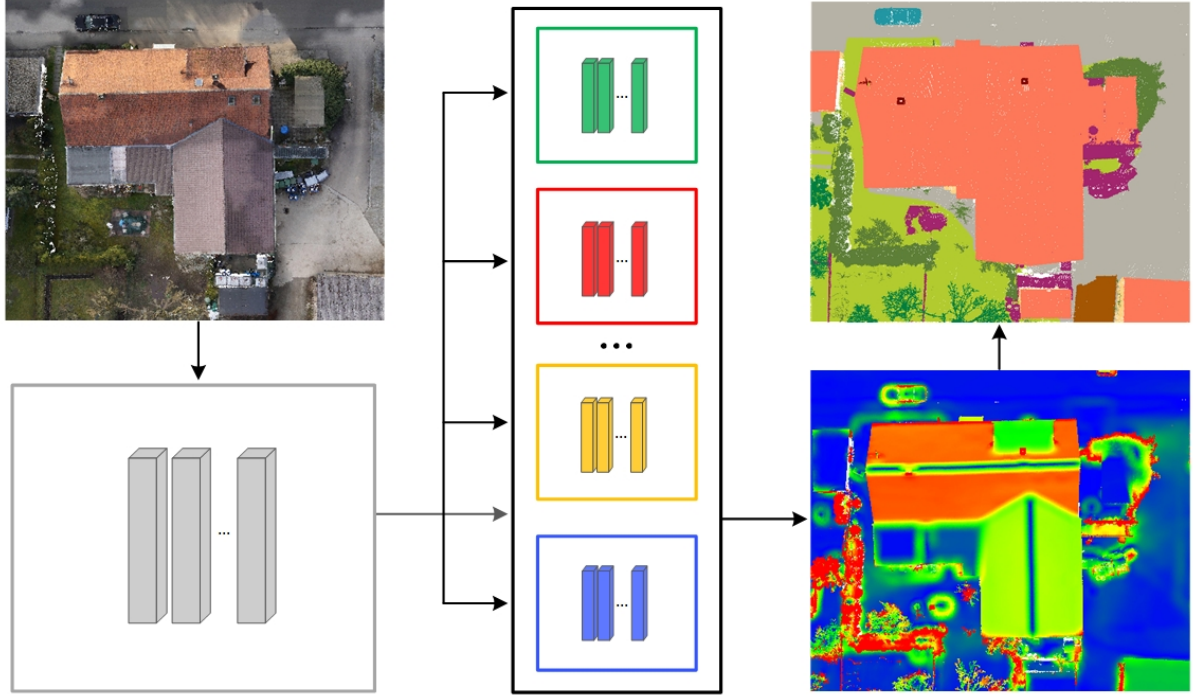


Figure 3.14: Workflow of target aggregators for rare classes. Boxes in different colors indicate distinct aggregators corresponding to different categories.

Specifically, for  $K$  rare classes, the local attention scores are separately weighted according to their respective confidence scores. The adjusted attention scores are then concatenated along the channel dimension, forming a feature vector of dimension  $C \times K$ , as illustrated in Fig. 3.14. The module highlighted with the blue box in this figure is also depicted in conjunction with Fig. 3.12. Subsequently, an MLP layer is applied to integrate the concatenated feature vector back into the original  $C$  dimensional feature. The whole process can be formulated as:

$$T = \tau(w_1(A^{N \times C}) \odot w_2(A^{N \times C}) \odot \dots \odot w_K(A^{N \times C})) \quad (3.22)$$

where  $w_K(\cdot)$  represents the weight operation based on confidence scores of the  $K_{th}$  rare class. The symbol  $\odot$  is the concatenation operation. It should be mentioned that the original local attention is concatenated with the adjusted scores as well. The function  $\tau(\cdot)$  denotes the embedding process implemented by a linear layer, a batch normalization layer, and a ReLU activation layer. The output  $T \in \mathbb{R}^{N \times C}$  represents the rare target-aware attention scores, which can enhance the model's ability to capture and represent the nuanced spatial relationships inherent in rare classes and complex scenes.

### 3.4.7 Details of the network architecture

The designed modules from our attention network in Section 3.1 are incorporated to construct the backbone, aiming to effectively aggregate local point attentions across layers. Subsequently, the target-aware attention scores are seamlessly integrated into the backbone network to form an end-to-end structure. The detailed network architecture is illustrated in Fig. 3.15, which strengthens the model's ability to focus on rare categories while preserving the local focus attention of spatial discrepancy. By harmonizing global contextual information with local geometric details and instance-level structural characteristics, the network can achieve robust segmentation performance on both majority and rare classes in large-scale point clouds.

The backbone adopts a U-Net-style structure that connects the local focus attention modules and the sampling modules through a hierarchical encoder-decoder framework. Specifically, the topological relationship constructed by neighboring points, i.e., KNNs searching, is embedded by the local focus module to gather point-wise attention scores. The farthest point sampling (FPS) and the trilinear interpolation serve as the down-sampling and up-sampling modules, respectively, to maintain the integrity of the original structure. We adopt a fourfold sampling ratio. Furthermore, an MLP is utilized after each down-sampling module and before each up-sampling module to embed features for the subsequent block. Skip connections enable efficient transfer of shallow and deep semantic features across network layers, thereby maintaining point-level attentions to detail. We utilize addition operations. It should be mentioned that as the network depth increases, we progressively add the number of local attention modules in the encoders while omitting them in the decoders. This design allows for the extraction of deep semantic information while maintaining computational efficiency and reserving sufficient resources for the target-aware aggregator. Following the backbone network, point-wise attention scores are first encoded into a compact representation by the TFE module and subsequently passed to the CAM module for candidate center selection. The TTP module then parses the structural characteristics of rare targets, while the TCC module refines the confidence scores associated with the target centers. Finally, the TFA module integrates

the target-aware attention scores with the local attention features through a weighted aggregation process, enhancing the model's sensitivity to rare classes.

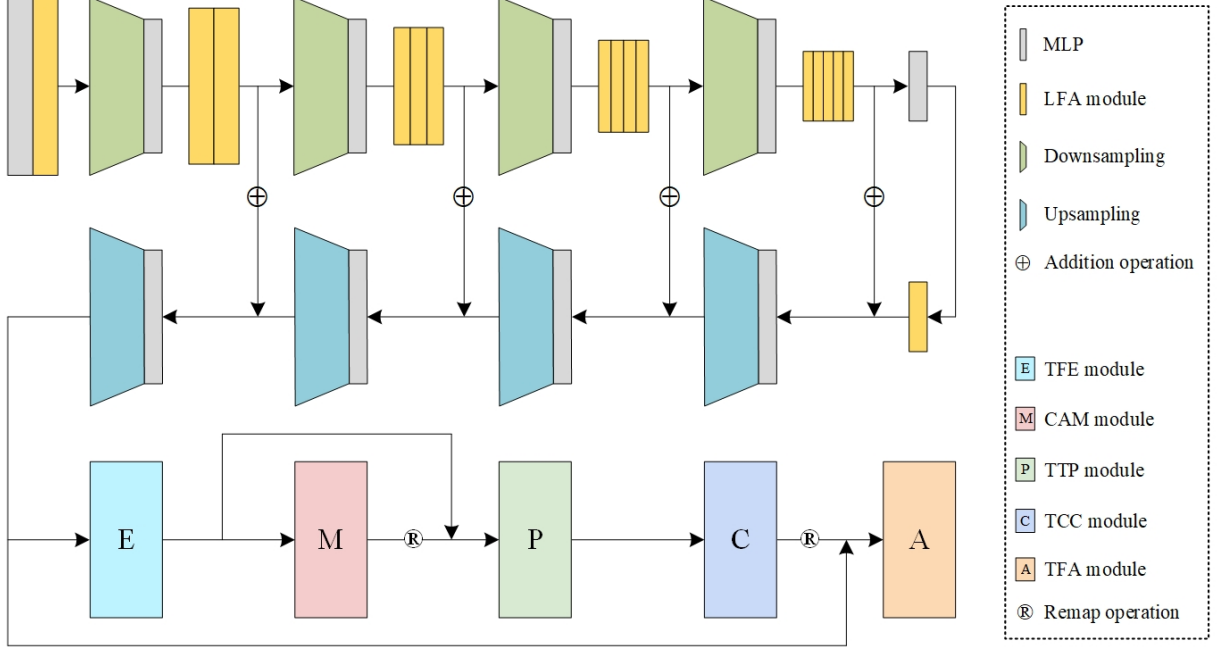


Figure 3.15: Detailed network architecture of the proposed target-aware attention. Local focus attention modules are integrated within a hierarchical framework to extract point-wise attentions. The target-aware aggregator produces structural features for rare classes and subsequently embeds them into the original 3D space.

For the semantic segmentation task, a fully connected layer is applied at the final stage to map the fused features into the label space, assigning each point to a specific semantic category. Given the class imbalance issue inherent in large-scale point clouds, the weighted cross-entropy loss is adopted for loss calculation. The loss function is defined as:

$$loss_{seg} = - \sum_{i=1}^N \sum_{j=1}^K \left( w_j \left( y_{ij} \log(p_{ij}) + (1 - y_{ij})(1 - \log(p_{ij})) \right) \right) \quad (3.23)$$

where  $N$  and  $K$  denote the total number of points and the total number of classes respectively. The term  $p_{ij}$  represents the predicted probability that the  $i_{th}$  point is assigned to the  $j_{th}$  class, while  $y_{ij} \in [0, 1]$  is the ground-truth label indicating whether the  $i_{th}$  point belongs to the  $j_{th}$  class. The weight  $w_j$  balances the importance of the  $j_{th}$  class, which is determined based on the statistical distribution of point categories. The entire network is trained end-to-end with a unified

loss function that combines the segmentation loss, the target parsing loss, and the candidate mask loss, as described in Section 3.4.3 and Section 3.4.4. The unified loss function is expressed as:

$$loss = loss_{seg} + \zeta \cdot loss_{ins} + \eta \cdot loss_{mask} \quad (3.24)$$

where  $\zeta$  and  $\eta$  are weighting coefficients used to balance the contributions of different tasks, effectively harmonizing the various losses during training and preventing any single task from dominating the optimization.



## Chapter 4

# Data & Evaluation Metrics

In this chapter, we will provide an overview of the datasets, including an analysis of their class distribution, and the evaluation metrics for the analysis of experimental results.

### 4.1 Dataset description

For the performance assessment of our method in the robustness to different data sources, experiments using three high-density urban point cloud benchmarks based on multiple platforms were conducted for evaluation and analysis. They are the airborne laser scanning (ALS) dataset Hessigheim 3D (Kölle et al., 2021), the mobile laser scanning (MLS) dataset Paris-Lille-3D (Roynard et al., 2018), the terrestrial laser scanning (TLS) dataset Semantic3D (Hackel et al., 2017), and the WHU-Urban3D dataset (Han et al., 2024) containing both MLS and ALS data.

#### Hessigheim 3D dataset

The dataset was captured by a Riegl VUX-1LR Scanner and two oblique-looking Sony Alpha 6000 cameras integrated on a RIEGL Ricopter platform in Hessigheim, Germany. It comprises a highly dense LiDAR point cloud around 800 points/m<sup>2</sup>, which is enriched by RGB images of the ground sampling distance of about 2-3 cm (see Fig. 4.1). The 9-dimensional input feature vector  $\{X, Y, Z, R, G, B, I, N_e, E_n\}$  consists of the coordinate, the texture color, the reflectance, the number of echoes and the echo number. The entire urban scene is subdivided into three distinct areas for training, validation, and testing. For efficient evaluation of our proposed method, this study employed training and validation sets with approximately 59.45 million and 14.46 million points, respectively. Eleven fine-grained classes have been manually annotated, including Low vegetation,

Impervious surface, Vehicle, Urban furniture, Roof, Facade, Shrub, Tree, Soil/Gravel, Vertical surface, and Chimney. The number of points varies considerably for different classes and the most underrepresented classes are Vehicle and Chimney.



Figure 4.1: Illustration of the Hessigheim 3D ALS dataset. The training set is colored according to available RGB data while the validation set is marked with gray shading.

| Class             |        | Low<br>Vege. | Impe.<br>Surf. | <u>Vehicle</u> | Urban<br>Furn. | Roof    | Facade  | Shrub   | Tree    | Soil/<br>Gravel | Vert.<br>Surf. | <u>Chimney</u> |
|-------------------|--------|--------------|----------------|----------------|----------------|---------|---------|---------|---------|-----------------|----------------|----------------|
| Traning<br>set    | points | 21375614     | 10419635       | 258032         | 1159205        | 6279431 | 1198227 | 1077141 | 8086818 | 8590706         | 974976         | 25321          |
|                   | ratio  | 35.96%       | 17.53%         | 0.43%          | 1.95%          | 10.56%  | 2.02%   | 1.81%   | 13.60%  | 14.45%          | 1.64%          | 0.04%          |
| Validation<br>set | points | 3738743      | 3212988        | 183263         | 455389         | 3052150 | 551996  | 341579  | 2218551 | 592510          | 101243         | 15836          |
|                   | ratio  | 25.85%       | 22.21%         | 1.27%          | 3.15%          | 21.10%  | 3.82%   | 2.36%   | 15.34%  | 4.10%           | 0.70%          | 0.11%          |

Table 4.1: Class frequencies in the training and validation sets for the Hessigheim 3D dataset. Note that the rare categories corresponding to each dataset are marked with underlined texts.

## Paris-Lille-3D Dataset

This dataset was collected with a vehicle-mounted MLS system, which was equipped with a GPS, an IMU and a Velodyne HDL-32E LiDAR. The dataset consists of three parts, i.e., two parts in the agglomeration of Lille and one in Paris, covering approximately 2 km of road segment with 140 million points, as illustrated in Fig. 4.2. The 4-dimensional input feature vector  $\{X, Y, Z, I\}$  is comprised of the coordinate and the reflectance. In this study, We extract the part Lille2 from the

training set as the validation set to efficiently validate our method. Nine categories were predefined for point cloud semantic segmentation task, including Ground, Building, Pole, Bollard, Trash can, Barrier, Pedestrian, Car, and Natural. Unclassified points without annotated labels in the dataset were excluded before training. The large variation of class distributions occurs on the dataset and the most imbalanced classes are Pole, Bollard, Trash can and Pedestrian.

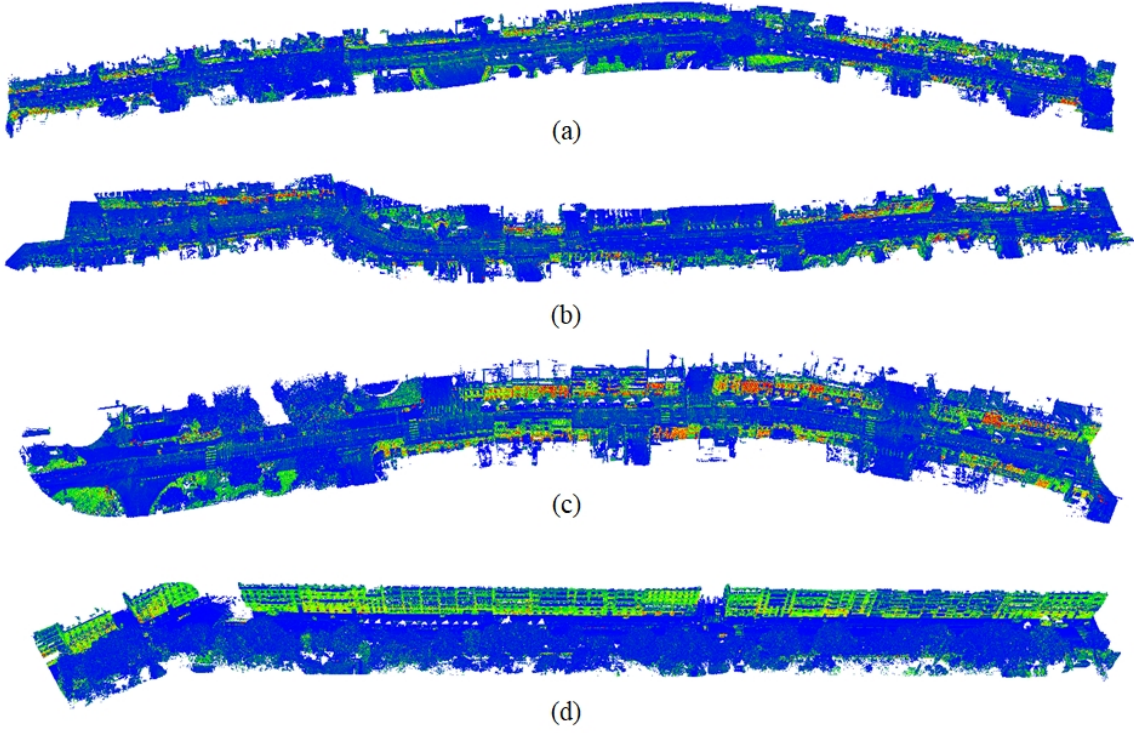


Figure 4.2: Illustration of the Paris-Lille-3D MLS dataset. All point clouds are colored with intensity values.

| Class          |        | Ground   | Building | <u>Pole</u> | <u>Bollard</u> | <u>Trash Can</u> | Barrier | <u>Pedestrian</u> | Car     | Natural |
|----------------|--------|----------|----------|-------------|----------------|------------------|---------|-------------------|---------|---------|
| Traning set    | points | 53033221 | 25821560 | 546190      | 54417          | 180638           | 3998129 | 160899            | 3441844 | 9681606 |
|                | ratio  | 54.72%   | 26.64%   | 0.56%       | 0.06%          | 0.19%            | 4.13%   | 0.17%             | 3.55%   | 9.99%   |
| Validation set | points | 12042099 | 7193259  | 109906      | 7298           | 115885           | 54818   | 11233             | 770451  | 917631  |
|                | ratio  | 56.74%   | 33.89%   | 0.52%       | 0.03%          | 0.55%            | 0.26%   | 0.05%             | 3.63%   | 4.32%   |

Table 4.2: Class frequencies in the training and validation sets for the Paris-Lille-3D dataset. Note that the rare categories corresponding to each dataset are marked with underlined texts.

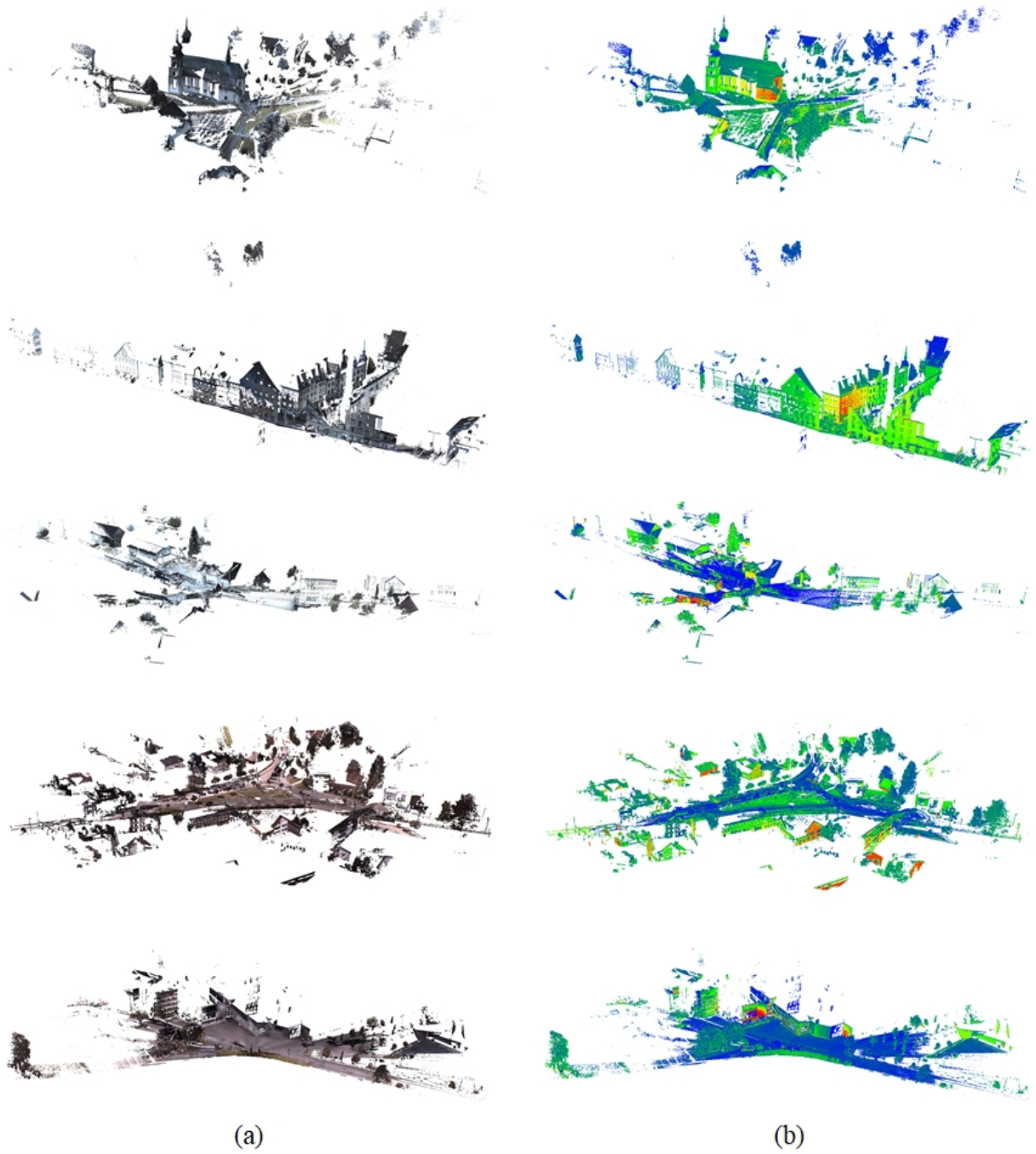


Figure 4.3: Illustration of the selected scenes from the Semantic3D TLS dataset. (a) shows the point clouds colored with RGB data; (b) illustrated the point clouds colored with intensity values.

## Semantic3D Dataset

The dataset was acquired by surveying-grade TLS platform in Central Europe, which consists of in total approximately 4 billions 3D points of much higher point density compared to the other datasets. Colorization was performed in a post processing step by deploying a high resolution cube-map generated from camera images. The 7-dimensional input feature vector  $\{X, Y, Z, I, R, G, B\}$  is composed of the coordinate, the reflectance, and the texture color. In this study, we selected 5 scenes from the original 15 training scenes for testing, i.e., bildstein5, domfountain3, neugasse, sg27\_9 and untermaederbrunnen3 (see Fig. 4.3). Eight classes were labeled for the purpose of semantic segmentation, including Man-made terrain, Natural terrain, High vegetation, Low vegetation, Buildings, Hard scape, Scanning artefacts, and Cars. Again, unlabeled points without predefined labels in the dataset were excluded before training. Most difficult classes, Scanning artefacts and Cars, have only few training and test samples and a large variation of possible object shapes.

| Class      |        | MM.<br>Terrain | Natu.<br>Terrain | High<br>Vege. | Low<br>Vege. | Buildings | Hard<br>Scape | Scan.<br>Arte. | <u>Cars</u> |
|------------|--------|----------------|------------------|---------------|--------------|-----------|---------------|----------------|-------------|
| Traning    | points | 586041101      | 473196629        | 131551840     | 98105678     | 213843372 | 72391435      | 50140299       | 6731528     |
| set        | ratio  | 35.91%         | 28.99%           | 8.06%         | 6.01%        | 13.10%    | 4.44%         | 3.07%          | 0.41%       |
| Validation | points | 35486486       | 4538592          | 3544846       | 1522438      | 52322820  | 4099549       | 1660838        | 2399245     |
| set        | ratio  | 33.61%         | 4.30%            | 3.36%         | 1.44%        | 49.56%    | 3.88%         | 1.57%          | 2.27%       |

Table 4.3: Class frequencies in the training and validation sets for the Semantic3D dataset. Note that the rare categories corresponding to each dataset are marked with underlined texts.

## WHU-Urban3D Dataset

This dataset was collected using the AS-900HL multi-platform LiDAR measurement system and the HiScan-Z 3D laser mobile measurement system. It provides three sub-datasets containing densely distributed and highly accurate 3D points from distinct urban areas in Shanghai and Wuhan, China. The MLS-S dataset encompasses more than 10 kilometers of urban roads, while the ALS-S dataset covers over 3.6 million square meters of urban scenes in Shanghai, as shown in Fig. 4.4. The MLS-W dataset includes over 4 kilometers of urban roads in Wuhan and serves as test data for evaluating cross-city generalizability. The 5-dimensional input feature vector of the MLS data,  $\{X, Y, Z, I, N_e\}$ , consists of the coordinate, the reflectance, and the number of echoes, while the 3-dimensional input feature vector of the ALS data,  $\{X, Y, Z\}$ , contains only coordinate information. Eighteen categories were annotated for the MLS data and seven categories were annotated for the ALS data. The minority classes defined by the WHU-Urban3D benchmark, including Traffic light, Box, Detector, and Municipal pole in the MLS data, and Light in the ALS data, challenge the model's ability to manage imbalanced classes.

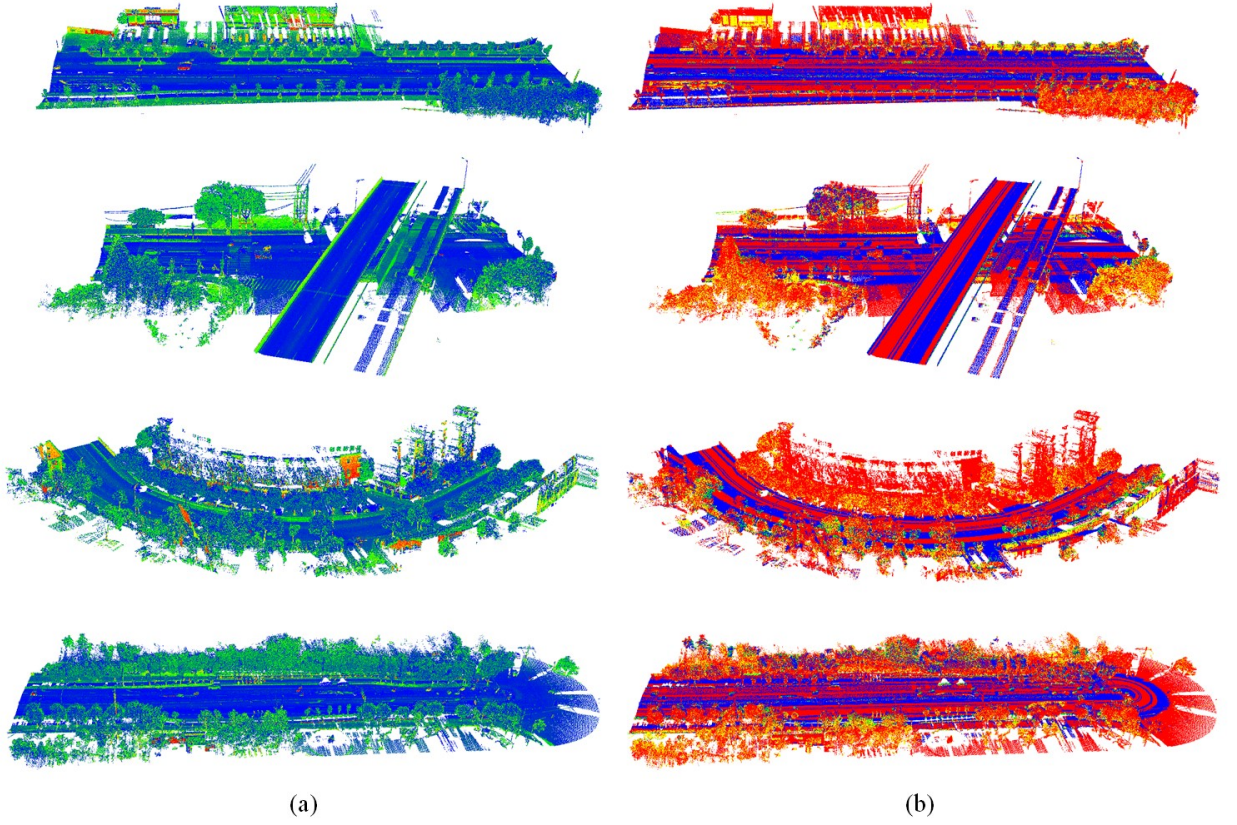


Figure 4.4: Illustration of the selected scenes from WHU-Urban3D MLS dataset. (a) shows the point clouds colored with RGB data; (b) presents the point clouds colored with number of echoes.

In case of large-scale point clouds, a significant imbalance in class distribution frequently emerges. This disparity arises because the relative number of points for classes is contingent on the physical area encompassed by respective objects. Consequently, small target objects contribute an exceedingly minimal proportion of the total points within datasets. According to the default setting of the WHU-Urban3D dataset, Traffic light, Box, Detector, and Municipal pole in the MLS data, and Light in the ALS data are considered minority classes. Detailed statistics of class occurrences in the other three datasets are shown in Tables 4.1, 4.2, and 4.3, where the number of points for different classes considerably varied. Classes with underlined texts exhibit significant imbalance in the datasets, which makes the rare class segmentation a challenging task.

## 4.2 Evaluation criteria

We utilized per-class F1 score ( $F1$ ), overall accuracy ( $OA$ ), mean F1 score ( $mF1$ ) of all classes, and mean F1 score of rare classes ( $rmF1$ ) to assess the performance of our proposed method. Among

these metrics, F1 score is specifically used to evaluate the classification performance on each single class, which can be expressed as follows:

$$\begin{aligned} PR_k &= \frac{TP_k}{TP_k + FP_k} \\ RC_k &= \frac{TP_k}{TP_k + FN_k} \\ F1_k &= \frac{2 \cdot PR_k \cdot RC_k}{PR_k + RC_k} \end{aligned} \tag{4.1}$$

where  $PR_k$ ,  $RC_k$  and  $F1_k$  denote precision, recall and F1 score of the class  $k$  receptively.  $TP_k$ ,  $FP_k$  and  $FN_k$  are true positives, false positives, and false negatives determined for the corresponding class in a confusion matrix. Whereas  $OA$  and  $mF1$  score are applied to assess the performance on the whole test dataset. Specifically,  $OA$  represents the percentage of correctly classified points in the total number of test points and  $mF1$  score stands for the harmonic mean value combining precision and recall of all categories.

All methods proposed in Chapter 3 are evaluated using the aforementioned standard metrics to ensure the comparability and comprehensiveness of the experimental results. However, since the methods presented in Section 3.3 and Section 3.4 focus on rare classes, the mean F1 score of rare classes ( $rmF1$ ) is specifically considered as a significant indicator for assessing segmentation performance on imbalanced classes. Meanwhile, to align with the predefined evaluation criteria for different datasets, the Intersection over Union (IoU) results are provided for a more comprehensive assessment in Appendix.



## Chapter 5

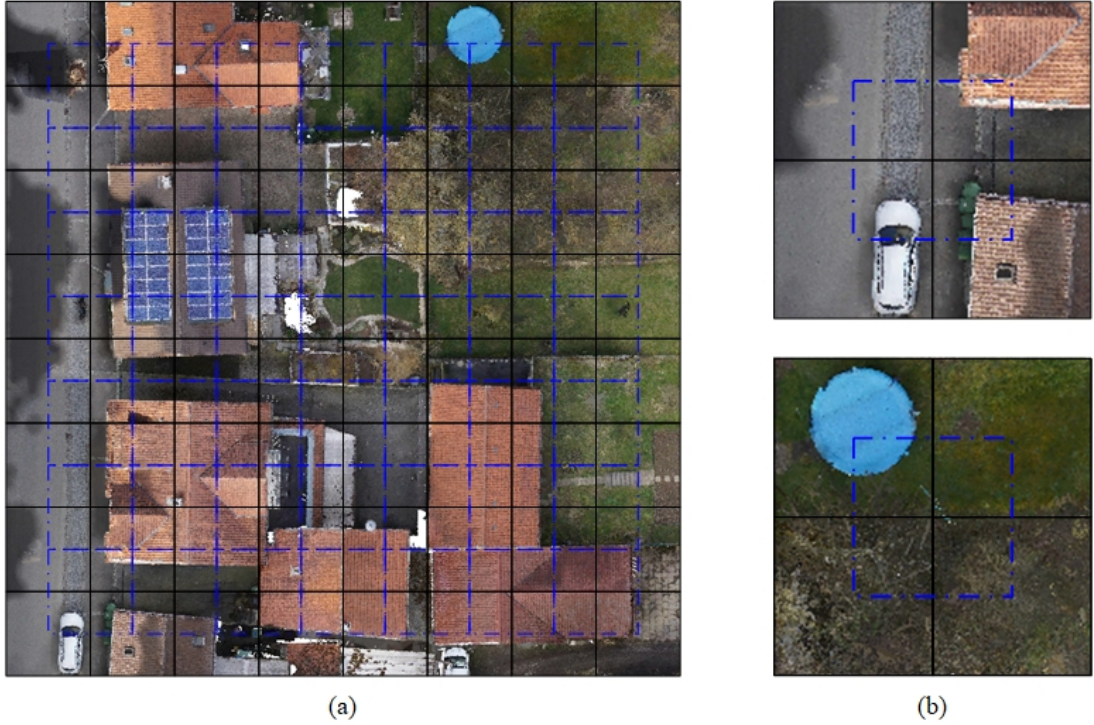
# Experiments & Results

In this chapter, we will introduce the experiment design, implementation details of preprocessing, experimental results, and method analysis. First, we provide a detailed introduction to the experimental design, which is based on the methodology outlined in Chapter 3 and the dataset overview in Chapter 4. The purpose is to address the challenges of large-scale urban point cloud semantic segmentation by proposing effective solutions, with particular emphasis on strategies for fine-grained object segmentation and rare class segmentation. Then, we describe the implementation details involved in data preparation, experimental hardware, and runtime settings. Next, we present the qualitative and quantitative experimental results of the proposed algorithms and methods listed in Chapters 3. Finally, we analyze these methods from different aspects based on the experimental results, providing a comprehensive evaluation of their performance and applicability.

### 5.1 Experiment design

For testing the performance of the methods introduced in Chapter 3, comprehensive experiments were conducted on multiple benchmark datasets to assess their effectiveness and robustness across various environments. First, we validate the feasibility of the attention network for large-scale point cloud introduced in Section 3.1 using datasets from different platforms, including the Hessigheim 3D dataset and the Paris-Lille-3D dataset. A systematic analysis is performed to examine the impact of different input features and module configurations on model performance. Next, based on the experimental results from the Hessigheim 3D dataset and the Paris-Lille-3D dataset, we further investigate the role of the local context embedding module presented in Section 3.2 in fine-grained segmentation tasks, evaluating its ability to capture intricate local geometric details. Following this, we test the contextual joint augmentation strategy described in Section 3.3 on the Hessigheim 3D

dataset and the Paris-Lille-3D dataset, analyzing its effectiveness in enhancing the diversity of rare category samples and improving model robustness. Finally, to verify the target-aware attention mechanism for rare class segmentation proposed in Section 3.4, we not only assess the model’s robustness on the two aforementioned datasets and the Semantic3D dataset, but also extend experiments on the WHU-Urban3D dataset to evaluate its generalization capability across diverse scenarios. Through this series of designed experiments, we comprehensively assess the effectiveness of the proposed methods, analyze the contributions of different modules to large-scale point cloud semantic segmentation, and validate their adaptability to different urban environments.



*Figure 5.1: Illustration of the training batch generation on the Hessigheim 3D dataset. (a) depicts the division of the training point cloud from top view, and the square frames in black and blue represent batch coverage areas with a 50% overlap. (b) shows the pattern connectivity of the example targets (vehicle and swimming pool) in detail.*

## 5.2 Implementation details

Considering the high density of the three benchmark datasets, we subsampled the raw training data with the grid size of 0.03 m during the preprocessing stage to reduce the computational burden while preserving point cloud structural details. During the inference process, predictions of Hessigheim 3D, Paris-Lille-3D and WHU-Urban3D datasets are conducted on raw test data. For Semantic3D dataset, the uniform downsampling with a resolution of 0.01 m was performed on the

test set because of the extremely high point density, which keeps the consistency with the default setting of Semantic3D benchmark. Prior to the training of the deep neural network, the training scenes were subdivided into  $6\text{ m} \times 6\text{ m}$  sub-regions in the form of pillars for mini-batch generation. As illustrated in Fig. 5.1, subsets of point clouds with a 50% overlap in both x- and y-directions were utilized to maintain pattern connectivity between adjacent sub-regions. We randomly sampled 16384 points from each subset as a training batch. It should be mentioned that if the point quantity is less than the predefined amount in the current subset, we perform a sampling with replacement, while we employ a sampling without replacement otherwise. In the inference phase, sub-regions without overlapping were obtained and sampled through a similar method. During the training process, the Adam optimizer is employed in the network with an initial learning rate of 0.001. Moreover, the training process lasted for 50 epochs totally and the model weights are saved if the loss decreases. The training and testing of all models was implemented in the framework of PyTorch and performed on a single NVIDIA GeForce RTX3090 24 GB GPU.

### 5.3 Results and analysis

This section presents the experimental results and corresponding analysis of the proposed framework for urban point cloud semantic segmentation. As illustrated in Fig. 5.2, we first analyze the performance of the attention-based backbone (Section 5.3.1), which is designed to capture long-range dependencies and complex interactions between points in large-scale urban point clouds. Subsequently, the impact of the local context embedding module on enhancing fine-grained geometric representation is examined (Section 5.3.2). Urban scenes inherently suffer from class imbalance, posing a significant challenge for rare class segmentation. At the data level (Section 5.3.3), the effectiveness of the contextual joint augmentation strategy in mitigating this imbalance is evaluated. At the model level (Section 5.3.4), the effectiveness of the target-aware aggregator in refining class-specific feature aggregation is analyzed.

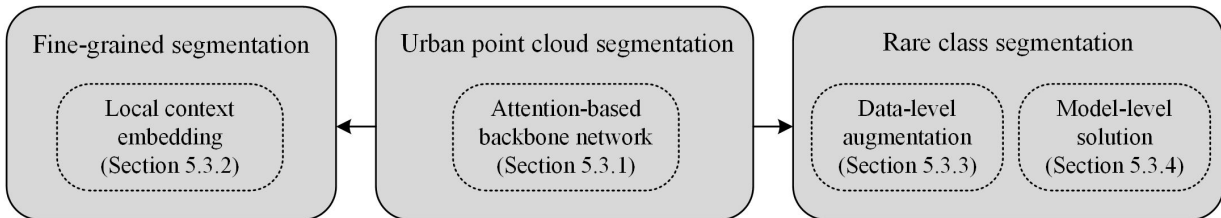


Figure 5.2: Analytical framework of the proposed method. The attention-based backbone serves as the core component of the framework. Fine-grained enhancement is achieved via local context embedding. Both data-level augmentation and model-level feature refinement are analyzed to address class imbalance.

### 5.3.1 Attention network for semantic segmentation of urban scenes

In this section, the performance of the proposed attention network is evaluated on large-scale urban datasets, including the Hessigheim 3D benchmark dataset and the Paris-Lille-3D benchmark dataset. The segmentation results of our attention network are compared with those of the classic convolutional method PointNet++, which is also a point-based deep learning method with a network architecture similar to our attention network. Furthermore, we also investigate the influence of multi-modality data input on our method, as well as the sensitivity of model performance to input noise and its computational complexity.

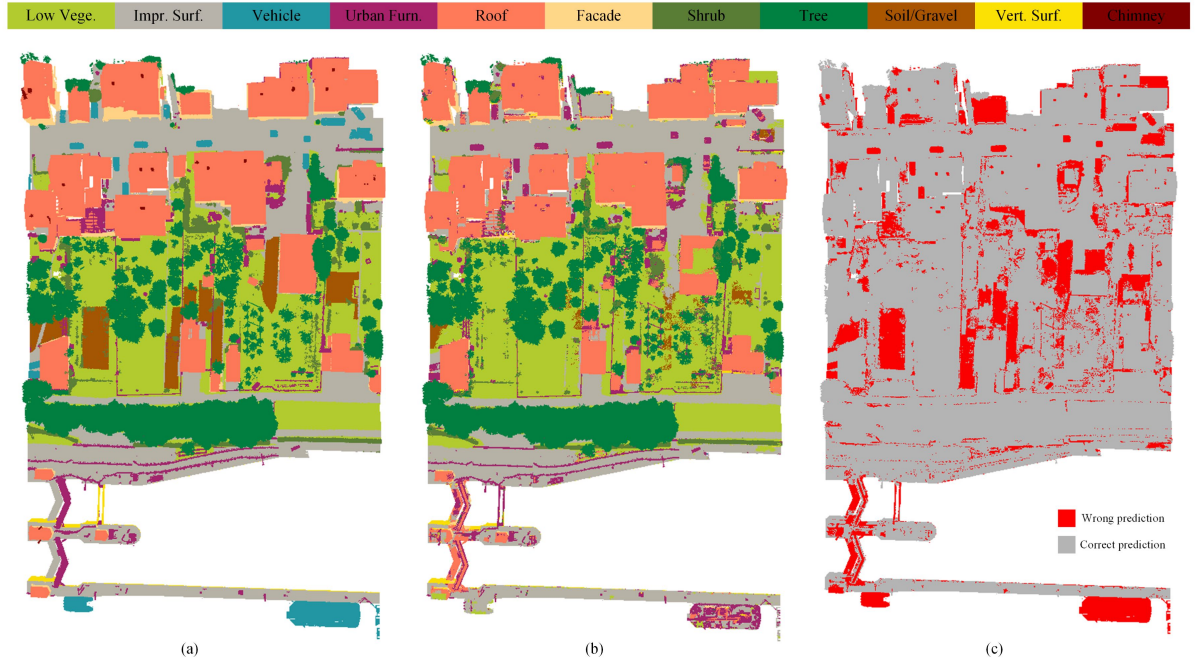


Figure 5.3: Classification results of the attention network on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

### Results analysis on the Hessigheim 3D dataset

To comprehensively evaluate the performance of the attention-based point cloud semantic segmentation network in complex urban scenarios, a systematic qualitative and quantitative analysis was conducted on the Hessigheim 3D dataset. As illustrated in Fig. 5.3 and Fig. 5.4, the segmentation results of the attention model are visually compared with their corresponding ground truth. The error map in Fig. 5.3(c) provides detailed insights into segmentation discrepancies. It can be clearly observed that the proposed attention-based network achieves high overall segmentation

quality, demonstrating excellent performance particularly in handling large-area, structurally stable categories such as roofs and trees. However, classification accuracy for fine-grained categories at object boundaries is somewhat limited, particularly in complex transitional regions with highly similar geometric structures, such as the interfaces between low vegetation and soil/gravel, where the segmentation performance still requires further improvement.

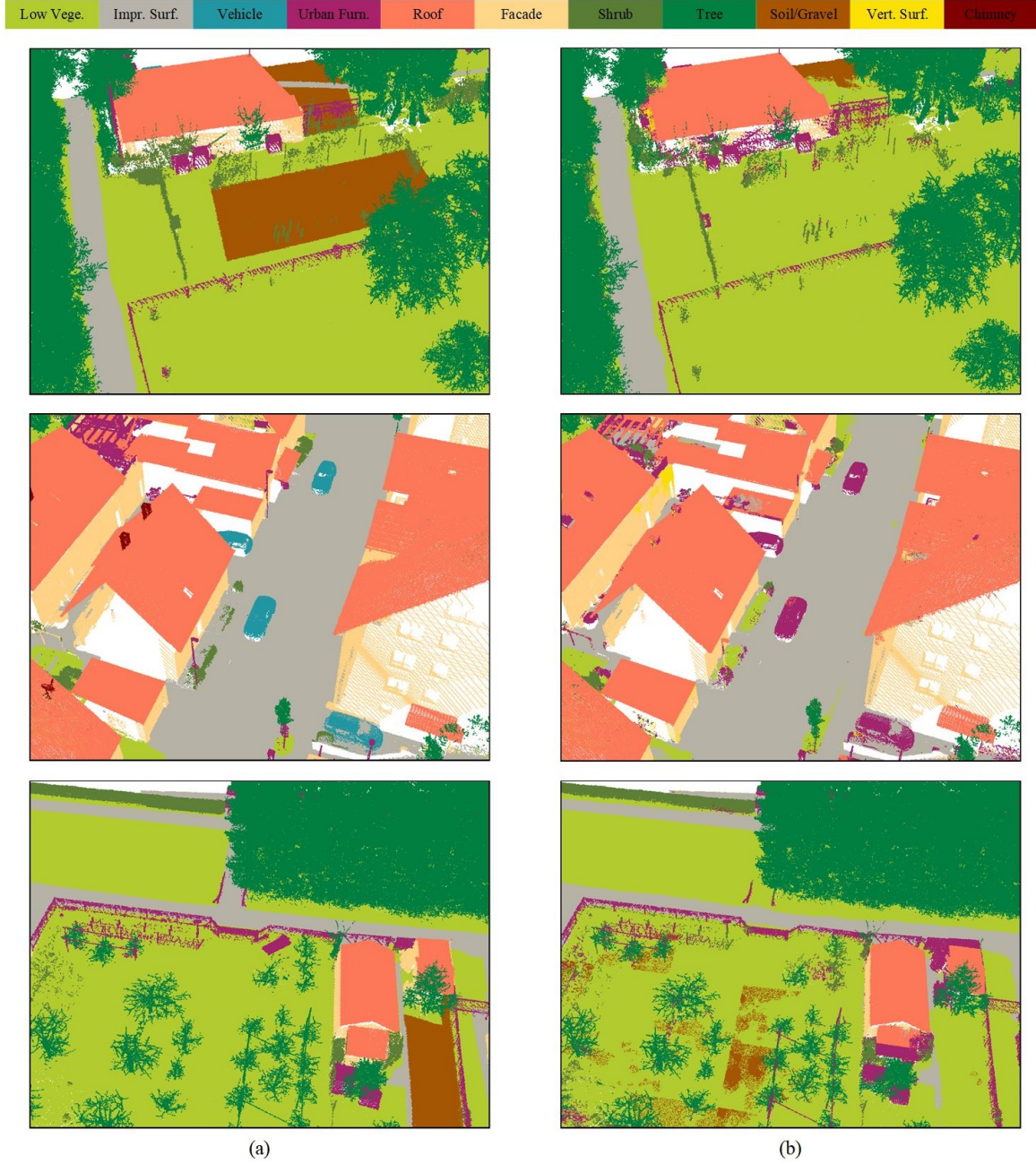


Figure 5.4: Detailed classification results of the attention network on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

To quantitatively assess the model’s segmentation performance, we further analyzed and compared the confusion matrices of our method and PointNet++. The detailed results are presented in Table 5.1 and Table 5.2. Our attention-based method achieved an OA of 83.8% and a mF1 of 57.8%. Compared to the convolution-based PointNet++, these results represent an improvement of 5.6% in OA and 11.7% in mF1, respectively. Furthermore, the experimental results demonstrate that the attention network exhibits superior segmentation performance in majority classes. For the tree category, the attention-based model achieved a Recall of 95.7%, a Precision of 94.6%, and an F1 score of 95.2%. This significantly outperforms the classic convolution-based method, PointNet++, which obtained an F1 score of 89.5% for the same class. Additionally, for large-area categories like roofs and impervious surfaces, the attention network achieved F1 scores of 92.8% and 87.5%, respectively, also significantly higher than PointNet++’s corresponding scores of 87.3% and 83.6%. This comprehensively demonstrates the significant effectiveness of the attention mechanism in processing large-scale, structurally distinct categories.

| Class       | Low<br>Vege. | Impe.<br>Surf. | Vehicle      | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | Chimney      | Precision |
|-------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|--------------|-----------|
| Low Vege.   | <b>0.907</b> | 0.092          | 0.087        | 0.098          | 0.027        | 0.036        | 0.107        | 0.015        | 0.601           | 0.008          | 0.001        | 0.793     |
| Impe. Surf. | 0.020        | <b>0.861</b>   | 0.144        | 0.062          | 0.040        | 0.061        | 0.006        | 0.000        | 0.086           | 0.013          | 0.000        | 0.891     |
| Vehicle     | 0.000        | 0.000          | <b>0.000</b> | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | 0.000        | 0.000     |
| Urban Furn. | 0.004        | 0.011          | 0.660        | <b>0.566</b>   | 0.022        | 0.067        | 0.059        | 0.004        | 0.004           | 0.091          | 0.315        | 0.444     |
| Roof        | 0.000        | 0.010          | 0.072        | 0.115          | <b>0.904</b> | 0.038        | 0.002        | 0.003        | 0.000           | 0.008          | 0.538        | 0.953     |
| Facade      | 0.001        | 0.002          | 0.028        | 0.061          | 0.003        | <b>0.694</b> | 0.005        | 0.001        | 0.000           | 0.116          | 0.077        | 0.850     |
| Shrub       | 0.008        | 0.000          | 0.000        | 0.050          | 0.000        | 0.013        | <b>0.544</b> | 0.020        | 0.000           | 0.000          | 0.001        | 0.644     |
| Tree        | 0.001        | 0.000          | 0.001        | 0.026          | 0.002        | 0.016        | 0.262        | <b>0.957</b> | 0.000           | 0.000          | 0.065        | 0.946     |
| Soil/Gravel | 0.060        | 0.017          | 0.003        | 0.001          | 0.000        | 0.000        | 0.000        | 0.000        | <b>0.309</b>    | 0.029          | 0.000        | 0.392     |
| Vert .Surf. | 0.001        | 0.007          | 0.006        | 0.022          | 0.002        | 0.076        | 0.015        | 0.001        | 0.000           | <b>0.735</b>   | 0.003        | 0.457     |
| Chimney     | 0.000        | 0.000          | 0.000        | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.000</b> | 0.000     |
| Recall      | 0.907        | 0.861          | 0.000        | 0.566          | 0.904        | 0.694        | 0.544        | 0.957        | 0.309           | 0.735          | 0.000        | 0.838     |
| F1          | 0.846        | 0.875          | 0.000        | 0.497          | 0.928        | 0.764        | 0.590        | 0.952        | 0.346           | 0.564          | 0.000        | 0.578     |

Table 5.1: Confusion matrix of the attention network on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label.

Despite the overall performance gains demonstrated by the confusion matrix analysis, certain limitations of the proposed model remain, particularly in handling fine-grained classes with complex spatial boundaries. For example, in the shrub category, our method achieves an F1 score of 59.0%, significantly outperforming PointNet++’s 25.8%. However, the performance in categories such as soil/gravel and urban furniture remains suboptimal, with F1 scores of only 34.6% and 49.7%, respectively. This indicates that these categories remain prominent bottlenecks for the model’s overall segmentation effectiveness. It can be suggested that, while the attention mechanism enhances the

model’s discriminative ability in regions with complex boundaries to a certain extent, the inherently ambiguous distribution and high geometric similarity of such categories make it challenging for the network to accurately capture fine-grained features in hierarchical feature extraction. Furthermore, for small-volume, sparsely distributed individual objects in the point cloud, the model also exhibits notable limitations. As shown in the confusion matrix in Table 5.1, the F1 scores for both the chimney and vehicle categories are 0.0%. This clearly indicates that, despite the attention mechanism contributing to overall performance, its ability to handle rare and sparsely distributed classes with limited samples remains significantly inadequate and requires further enhancement.

| Class       | Low<br>Vege. | Impe.<br>Surf. | Vehicle      | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | Chimney      | Precision |
|-------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|--------------|-----------|
| Low Vege.   | <b>0.931</b> | 0.176          | 0.164        | 0.376          | 0.131        | 0.113        | 0.327        | 0.017        | 0.774           | 0.007          | 0.093        | 0.654     |
| Impe. Surf. | 0.013        | <b>0.796</b>   | 0.181        | 0.055          | 0.045        | 0.060        | 0.004        | 0.000        | 0.109           | 0.022          | 0.024        | 0.881     |
| Vehicle     | 0.000        | 0.000          | <b>0.000</b> | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | 0.000        | 0.000     |
| Urban Furn. | 0.001        | 0.005          | 0.361        | <b>0.293</b>   | 0.009        | 0.037        | 0.011        | 0.000        | 0.000           | 0.092          | 0.263        | 0.473     |
| Roof        | 0.006        | 0.007          | 0.196        | 0.063          | <b>0.808</b> | 0.026        | 0.002        | 0.000        | 0.000           | 0.006          | 0.257        | 0.950     |
| Facade      | 0.001        | 0.002          | 0.059        | 0.035          | 0.003        | <b>0.665</b> | 0.006        | 0.001        | 0.000           | 0.630          | 0.274        | 0.765     |
| Shrub       | 0.001        | 0.000          | 0.001        | 0.019          | 0.000        | 0.006        | <b>0.159</b> | 0.004        | 0.000           | 0.001          | 0.003        | 0.679     |
| Tree        | 0.040        | 0.001          | 0.033        | 0.152          | 0.005        | 0.083        | 0.489        | <b>0.979</b> | 0.001           | 0.060          | 0.084        | 0.824     |
| Soil/Gravel | 0.008        | 0.012          | 0.001        | 0.001          | 0.000        | 0.000        | 0.001        | 0.000        | <b>0.116</b>    | 0.065          | 0.000        | 0.477     |
| Vert. Surf. | 0.000        | 0.002          | 0.005        | 0.007          | 0.000        | 0.009        | 0.002        | 0.000        | 0.000           | <b>0.120</b>   | 0.002        | 0.427     |
| Chimney     | 0.000        | 0.000          | 0.000        | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.000</b> | 0.000     |
| Recall      | 0.931        | 0.796          | 0.000        | 0.293          | 0.808        | 0.665        | 0.159        | 0.979        | 0.116           | 0.120          | 0.000        | 0.782     |
| F1          | 0.768        | 0.836          | 0.000        | 0.362          | 0.873        | 0.711        | 0.258        | 0.895        | 0.186           | 0.187          | 0.000        | 0.461     |

Table 5.2: Confusion matrix of PointNet++ on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label.

## Results analysis on the Paris-Lille-3D dataset

In addition to the experiments conducted on the ALS-based Hessigheim 3D dataset, we further evaluate our approach on the MLS dataset Paris-Lille-3D, in order to examine the adaptability and robustness of the proposed method across different sensing platforms and urban scenarios. To ensure consistency, we adopted the same evaluation framework used in the previous experiments. The pointwise semantic segmentation results are illustrated in Fig. 5.5, and the quantitative evaluations are summarized in Table 5.3 and Table 5.4. The visualization results of the segmentation outputs, shown in Fig. 5.5, reveals that our attention network produces highly consistent and semantically coherent results in diverse urban landscapes. The core urban classes, including buildings, roads, and natural vegetation, are accurately segmented with consistent boundaries, which demonstrates the robustness of the model in handling large continuous planar structures and irregular natural

forms. However, segmentation difficulties remain for fine objects, including bollards and barriers, which often appear sparsely and exhibit slender geometries.

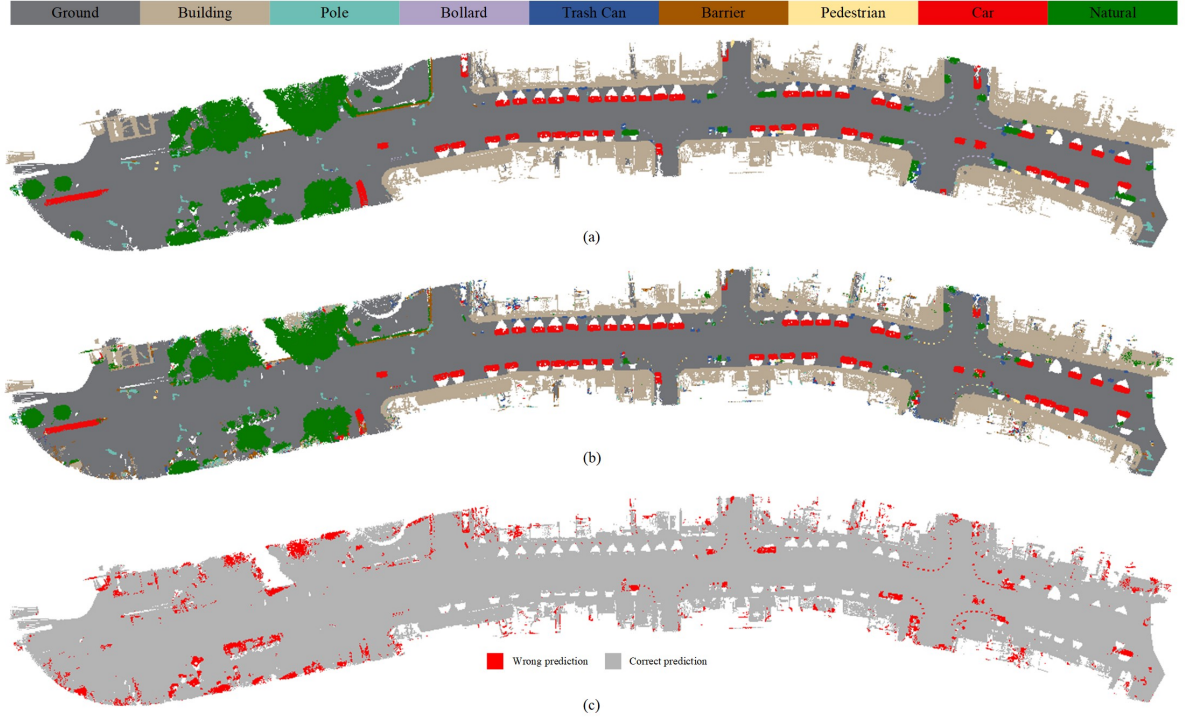


Figure 5.5: Classification results of the attention network on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

| Class      | Ground       | Building     | <u>Pole</u>  | <u>Bollard</u> | <u>Trash Can</u> | Barrier      | <u>Pedestrian</u> | Car          | Natural      | Precision |
|------------|--------------|--------------|--------------|----------------|------------------|--------------|-------------------|--------------|--------------|-----------|
| Ground     | <b>0.998</b> | 0.004        | 0.002        | 0.147          | 0.058            | 0.015        | 0.017             | 0.002        | 0.145        | 0.986     |
| Building   | 0.001        | <b>0.982</b> | 0.036        | 0.000          | 0.032            | 0.100        | 0.069             | 0.004        | 0.025        | 0.993     |
| Pole       | 0.000        | 0.001        | <b>0.869</b> | 0.199          | 0.002            | 0.108        | 0.032             | 0.000        | 0.010        | 0.803     |
| Bollard    | 0.000        | 0.000        | 0.000        | <b>0.000</b>   | 0.000            | 0.000        | 0.000             | 0.000        | 0.000        | 0.000     |
| Trash Can  | 0.000        | 0.002        | 0.000        | 0.000          | <b>0.734</b>     | 0.003        | 0.013             | 0.003        | 0.002        | 0.827     |
| Barrier    | 0.001        | 0.003        | 0.030        | 0.128          | 0.028            | <b>0.636</b> | 0.065             | 0.003        | 0.008        | 0.428     |
| Pedestrian | 0.000        | 0.000        | 0.001        | 0.462          | 0.009            | 0.020        | <b>0.641</b>      | 0.001        | 0.001        | 0.415     |
| Car        | 0.000        | 0.001        | 0.003        | 0.031          | 0.130            | 0.003        | 0.012             | <b>0.985</b> | 0.001        | 0.962     |
| Natural    | 0.001        | 0.007        | 0.058        | 0.034          | 0.007            | 0.116        | 0.152             | 0.002        | <b>0.807</b> | 0.908     |
| Recall     | 0.998        | 0.982        | 0.869        | 0.000          | 0.734            | 0.636        | 0.641             | 0.985        | 0.807        | 0.980     |
| F1         | 0.992        | 0.987        | 0.835        | 0.000          | 0.777            | 0.512        | 0.504             | 0.973        | 0.855        | 0.715     |

Table 5.3: Confusion matrix of the attention network on the Paris-Lille-3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

| Class      | Ground | Building | <u>Pole</u> | <u>Bollard</u> | <u>Trash Can</u> | Barrier | <u>Pedestrian</u> | Car   | Natural | Precision |
|------------|--------|----------|-------------|----------------|------------------|---------|-------------------|-------|---------|-----------|
| Ground     | 0.973  | 0.010    | 0.009       | 0.214          | 0.181            | 0.029   | 0.056             | 0.077 | 0.065   | 0.982     |
| Building   | 0.013  | 0.913    | 0.180       | 0.124          | 0.507            | 0.319   | 0.311             | 0.121 | 0.114   | 0.935     |
| Pole       | 0.000  | 0.001    | 0.475       | 0.015          | 0.004            | 0.087   | 0.001             | 0.002 | 0.001   | 0.804     |
| Bollard    | 0.000  | 0.000    | 0.000       | <b>0.000</b>   | 0.000            | 0.000   | 0.000             | 0.000 | 0.000   | 0.000     |
| Trash Can  | 0.000  | 0.000    | 0.000       | 0.000          | 0.000            | 0.000   | 0.000             | 0.000 | 0.000   | 0.000     |
| Barrier    | 0.001  | 0.008    | 0.103       | 0.154          | 0.085            | 0.215   | 0.066             | 0.072 | 0.021   | 0.064     |
| Pedestrian | 0.000  | 0.000    | 0.002       | 0.001          | 0.000            | 0.002   | 0.042             | 0.001 | 0.000   | 0.261     |
| Car        | 0.001  | 0.001    | 0.098       | 0.402          | 0.194            | 0.106   | 0.038             | 0.687 | 0.004   | 0.887     |
| Natural    | 0.012  | 0.067    | 0.133       | 0.090          | 0.029            | 0.243   | 0.487             | 0.041 | 0.795   | 0.512     |
| Recall     | 0.973  | 0.913    | 0.475       | 0.000          | 0.000            | 0.215   | 0.042             | 0.687 | 0.795   | 0.924     |
| F1         | 0.977  | 0.924    | 0.597       | 0.000          | 0.000            | 0.099   | 0.072             | 0.774 | 0.623   | 0.452     |

Table 5.4: Confusion matrix of PointNet++ on the Paris-Lille-3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

The quantitative results, reported in Table 5.3 and Table 5.4, further confirm the effectiveness of the attention-based method. The proposed network achieved an OA of 98.0% and a mF1 score of 71.5%, outperforming the baseline PointNet++ by 5.6% in OA and 26.3% in mF1. It is worth noting that the attention-based network not only surpasses PointNet++ on large-area categories such as ground, buildings, and natural vegetation, but also demonstrates remarkable improvements in the segmentation of small urban objects, including poles, trash cans, barriers, and pedestrians. These categories are typically more challenging due to their sparse geometric representation and frequent occlusions, especially when captured from the perspective of mobile laser scanning. Nevertheless, the segmentation of bollards remains a critical weakness. This limitation can be attributed to insufficient geometric distinctiveness, since bollards often resemble poles or barriers in shape and scale. Another contributing factor is the extremely sparse point coverage, which makes it difficult for the model to capture the discriminative features of bollards. It suggests that while the attention mechanism significantly improves the recognition of most small-scale categories, additional strategies such as the incorporation of local geometric context, the class-balanced augmentation, or target-aware feature extraction may be required to address the segmentation of extremely underrepresented classes.

### Influence of multi-modality input data

Leveraging the well-registered multi-sensor data provided by the Hessigheim 3D benchmark dataset, we conducted experiments to evaluate the impact of multi-modality inputs on semantic segmentation performance of the attention model. Considering the accessibility of open-source datasets,

the comparative analysis primarily focuses on the intensity features of LiDAR point clouds and the texture information derived from RGB images. Building upon the fundamental input of LiDAR point clouds, spatial coordinates and intensity values were concatenated with the RGB texture information to form the initial input to the segmentation network. Experimental results are shown in Table 5.5 and demonstrate that introducing the RGB texture significantly enhances segmentation performance. Compared with the baseline model A, the OA and the mF1 score of the model D improved by 4.2% and 4.5%, respectively. Specifically, F1 scores for classes such as low vegetation and impervious surfaces increased from 81.7% to 88.7% and from 84.5% to 89.6%, respectively. Among these, the soil/gravel class characterized by distinctive color patterns and rich intensity variations benefited the most from the integration of multi-modality inputs, achieving an F1 score improvement of 28.9%. These results indicate the effectiveness of multi-modality data fusion in addressing the limitations of pure geometry representations, particularly in capturing texture-specific details. However, it is important to note that in regions with complex lighting conditions or low color contrast, the inclusion of extra RGB information does not always yield consistent improvements and may even degrade performance for certain categories. For instance, the F1 score for the shrub category in the model C exhibited a decline of 2.8% relative to the model A. When compared with the model B, the model D demonstrated reductions in F1 scores of 7.2% and 3.7% for the vertical surface and soil/gravel categories, respectively. Consequently, a critical challenge in multi-modality fusion lies in balancing the contributions of different data sources to mitigate the potential noise introduced by additional channels. Future research may explore the incorporation of adaptive weighting mechanisms or dynamic feature selection strategies within the fusion framework to further optimize model performance, especially under challenging illumination conditions.

| Method | Input |     | F1 Score     |                |         |                |             |             |             |             |                 |                |         | OA          | mF1         |
|--------|-------|-----|--------------|----------------|---------|----------------|-------------|-------------|-------------|-------------|-----------------|----------------|---------|-------------|-------------|
|        | I     | RGB | Low<br>Vege. | Impe.<br>Surf. | Vehicle | Urban<br>Furn. | Roof        | Facade      | Shrub       | Tree        | Soil/<br>Gravel | Vert.<br>Surf. | Chimney |             |             |
| A      |       |     | 81.7         | 84.5           | 0.0     | 49.2           | <b>94.1</b> | 75.2        | 54.9        | 94.8        | 3.7             | 56.6           | 0.0     | 81.9        | 54.1        |
| B      | ✓     |     | 84.6         | 87.5           | 0.0     | 49.7           | 92.8        | 76.4        | <b>59.0</b> | 95.2        | 34.6            | 56.4           | 0.0     | 83.8        | 57.8        |
| C      |       | ✓   | 88.5         | 89.2           | 0.0     | <b>50.8</b>    | 93.6        | <b>77.8</b> | 52.1        | 94.3        | <b>36.2</b>     | <b>66.0</b>    | 0.0     | 86.0        | <b>59.0</b> |
| D      | ✓     | ✓   | <b>88.7</b>  | <b>89.6</b>    | 0.0     | 49.0           | 93.8        | 77.8        | <b>59.0</b> | <b>95.3</b> | 32.6            | 58.8           | 0.0     | <b>86.1</b> | 58.6        |

Table 5.5: Comparison of the semantic segmentation results on the Hessigheim 3D dataset using the attention network with different modality inputs (values in %). Note that the highest values in per-class F1 score, OA, and mF1 are highlighted in bold text.

## Sensitivity analysis of input noise

In practical applications, point cloud data unavoidably suffers from sensor noise and environmental interference, resulting in a certain degree of noise within the data. To comprehensively evaluate the robustness of the attention model, we introduce Gaussian noise with varying intensities and

proportions into the original point cloud data. Experimental results on the Hessigheim 3D dataset in Fig. 5.6(a) demonstrate that as the noise intensity increases, the OA of conventional convolution-based PointNet++ model degrades significantly, whereas our attention-based network exhibits a considerably smaller performance drop. Specifically, when the standard deviation of the noise is set to 0.005, the OA of PointNet++ decreases by 7.6%, while the OA of the attention-based network drops by only 3.1%. Even when the noise standard deviation increases to 0.01, the OA of PointNet++ declines by 16.9%, whereas the attention-based model only drops by 12.1%. These results clearly indicate the superior robustness of the attention mechanism under noisy conditions compared to PointNet++. Furthermore, as shown in Fig. 5.6(b), when the standard deviation is fixed at 0.005 and the proportion of noise is progressively increased, the OA of PointNet++ undergoes a steep decline, whereas the OA of the attention network remains relatively stable. This difference in robustness can be attributed to the inherent advantages of attention mechanisms in facilitating global feature interaction. By enhancing information exchange between distant point pairs within the point cloud, the attention mechanism remains effective in capturing critical global structural features even when local features are disrupted by noise. Maintaining the stability of global features is critical for improving the overall consistency and noise resilience of segmentation results. Therefore, the proposed method exhibits remarkable robustness under substantial noise interference, highlighting its strong potential for deployment in complex urban environments.

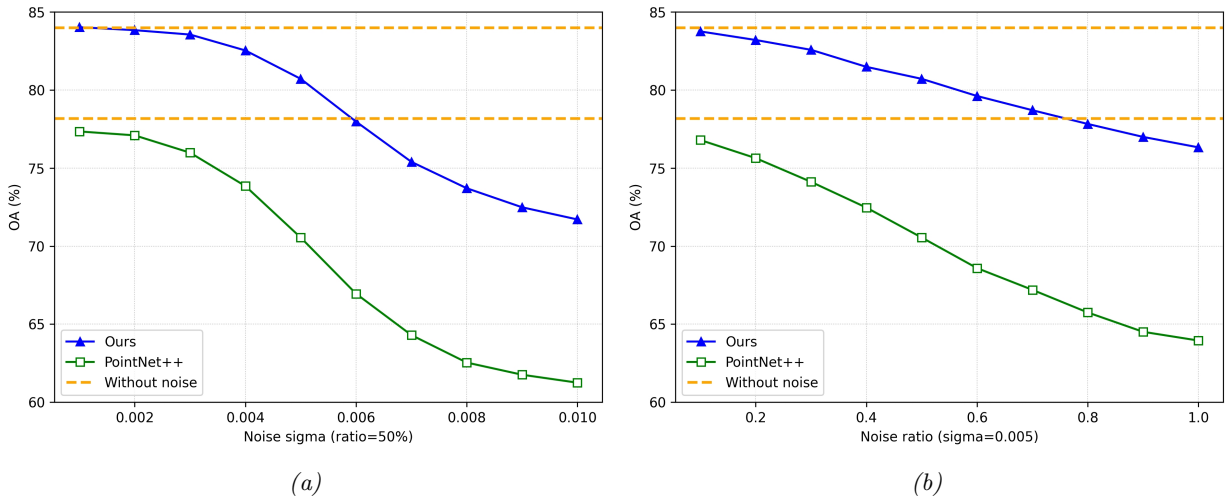


Figure 5.6: Sensitivity analysis of input noise for the attention network.

## Complexity and runtime analysis

Beyond accuracy and robustness, model complexity and operational efficiency are also crucial factors in evaluating a model’s practicality in real-world applications. We quantified the number of parameters, training time, and GPU memory usage of the attention-based network and compared it with PointNet++. The results on the Hessigheim 3D dataset are summarized in Table 5.6. Specifically, the number of parameters reflects the overall complexity of the model, which directly affects the running time and memory usage. The running time indicates the computational efficiency, while the GPU memory usage represents the hardware resources required to run the model. Comparative results indicate that despite a significant increase in the proposed model’s parameter count relative to PointNet++ due to the attention mechanism, its overall complexity remains moderate in terms of parameter scale, especially when considering the rapid advancements in hardware. Regarding the operational efficiency, the attention mechanism needs to calculate the pairwise point correlations. This introduces additional computational overhead compared to convolution networks, which rely on shared-weight local computations, leading to a longer running time per iteration and higher GPU memory consumption. Nevertheless, considering the powerful global feature capture capabilities and the significant improvement in final segmentation performance achieved by the attention mechanism, these modest increases in computational overhead are entirely acceptable.

| Method     | Params<br>(millions) | Running time<br>(ms/iter) | Memory<br>(GB) |
|------------|----------------------|---------------------------|----------------|
| PointNet++ | 0.96                 | 238                       | 1.39           |
| Ours       | 11.55                | 511                       | 8.16           |

Table 5.6: Comparison of number of parameters, computation time, and GPU memory usage of PointNet++ and our attention network. It is estimated and averaged on the Hessigheim 3D dataset.

### 5.3.2 Local context embedding for fine-grained segmentation

This section evaluates the proposed local context embedding module for fine-grained segmentation on two benchmark datasets, including the Hessigheim 3D dataset and the Paris-Lille-3D dataset. The results are compared with those of the baseline model presented in Section 5.3.1. In addition, we compare the proposed feature-level fusion strategy with input-level hard fusion, analyze the impact of different context scales by contrasting multi-scale and single-scale embeddings, and conclude with a complexity analysis of the model.

## Results analysis on the Hessigheim 3D dataset

To validate the effectiveness of the proposed SCE module in complex urban point cloud scenarios, we conducted both qualitative and quantitative evaluations using the Hessigheim 3D dataset. The focus was to assess the performance improvement in fine-grained semantic segmentation tasks. Specifically, we compared the segmentation results of a baseline attention based network with those obtained after incorporating the SCE module. The segmentation results, as illustrated in Fig. 5.7 and Fig. 5.8, demonstrate that the model maintains high overall segmentation quality after integrating the SCE module. In comparison with the baseline model without the SCE module in Fig. 5.3, the spatial coherence and consistency of the classification outputs are significantly improved. The edges of building roofs are more sharply delineated, particularly in architecturally complex building clusters. Moreover, category confusion between similar ground-level classes, such as soil/gravel and low vegetation, is considerably reduced. Visualization of the error map in Fig. 5.7(c) further highlights the enhanced segmentation accuracy along fine-grained structural boundaries. Issues such as blurred boundaries and inter-class ambiguity are substantially mitigated.

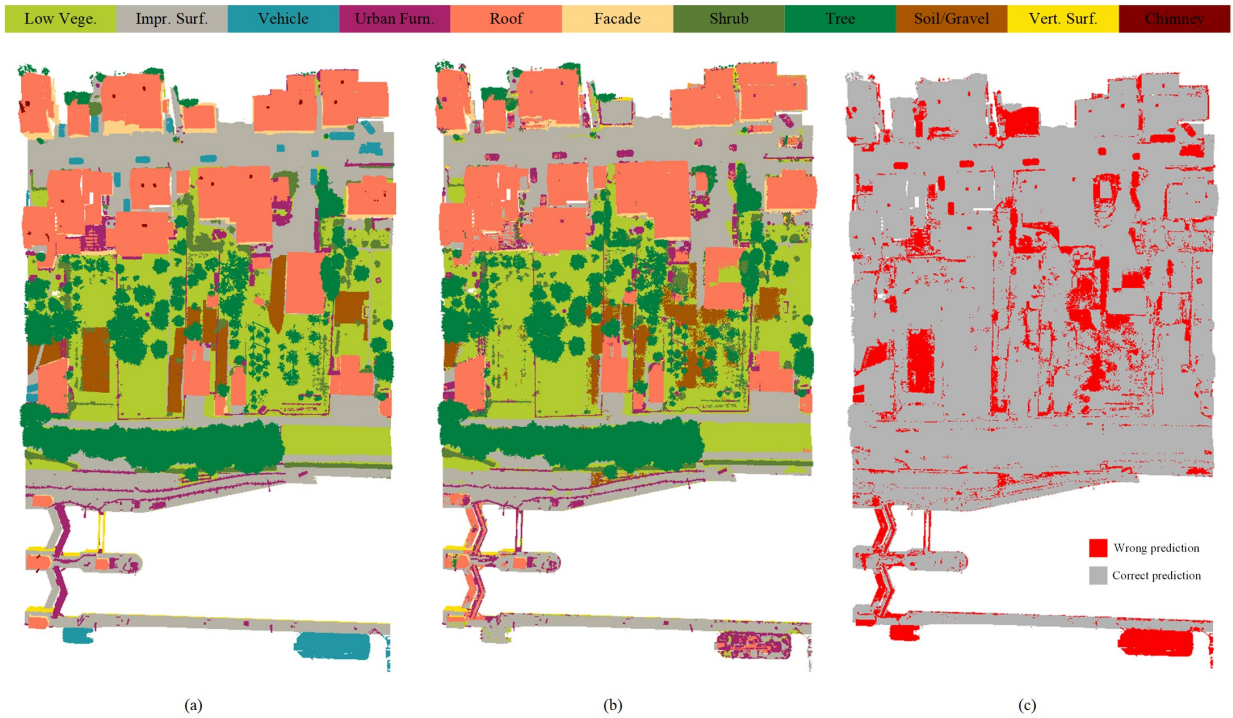


Figure 5.7: Classification results using proposed SCE module on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

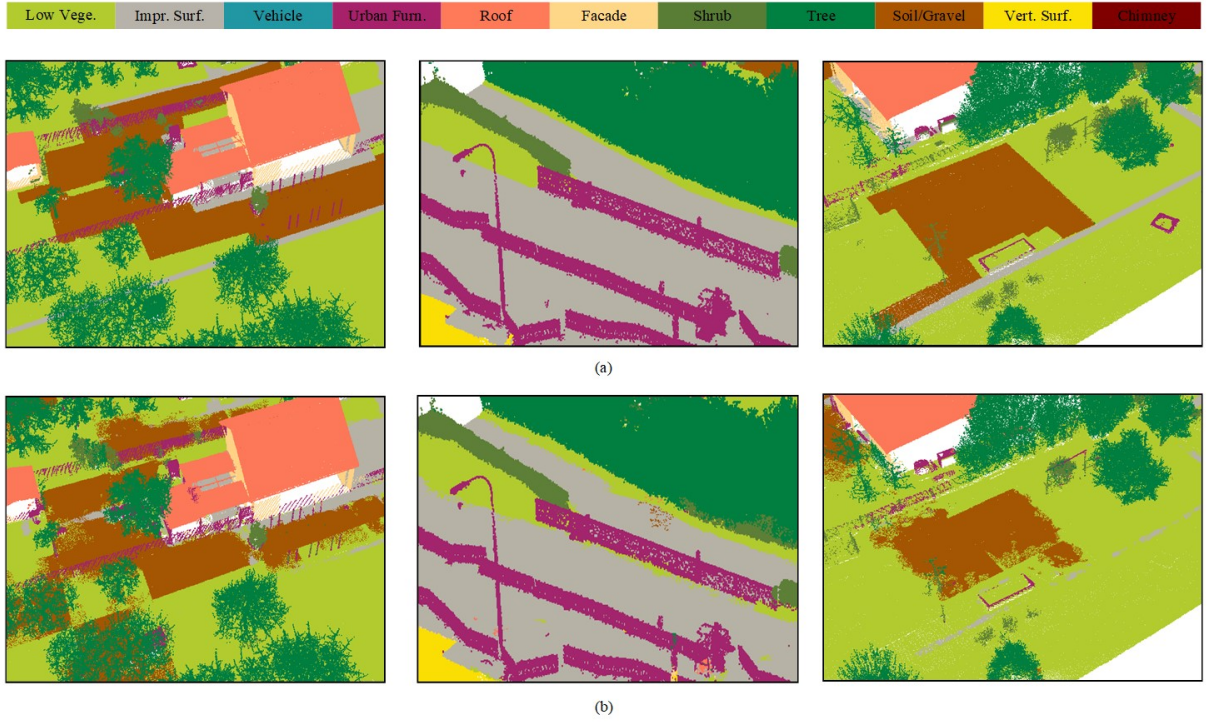


Figure 5.8: Detailed classification results using proposed SCE module on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

The confusion matrix in Table 5.7 provides further quantitative support for the effectiveness of the SCE module. When compared with the confusion matrix of the baseline method in Table 5.1, the mF1 score improves from 57.8% to 60.4%, while the OA increases from 83.8% to 85.4%. The results clearly indicate that the introduction of the SCE module leads to a notable enhancement in segmentation performance over the standard attention network alone. Notably, the proposed method achieves significant gains in distinguishing classes with ambiguous boundaries or similar geometric structures. For example, the F1 score for the soil/gravel class increases from 34.6% to 47.3%, and for the vertical facade class, it improves from 56.4% to 63.6%. Such improvements are attributed to the ability of the designed SCE module to enrich the representation of local neighborhood features during encoding, which strengthens the network perception of fine-grained elements in complex urban scenes. However, challenges remain in recognizing small-scale, sparsely distributed rare object classes, such as vehicle and chimney. The difficulties likely arise from the limited number of points associated with these objects and the lack of robust local contextual features. This scarcity hinders the ability of the network to learn stable and discriminative feature representations for these underrepresented classes. Future work could address these limitations through customized data augmentation strategies and target-aware perception mechanisms to enhance the segmentation performance for such challenging rare classes.

| Class       | Low<br>Vege. | Impe.<br>Surf. | Vehicle      | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | Chimney      | Precision |
|-------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|--------------|-----------|
| Low Vege.   | <b>0.895</b> | 0.084          | 0.060        | 0.089          | 0.013        | 0.030        | 0.092        | 0.012        | 0.445           | 0.047          | 0.000        | 0.826     |
| Impe. Surf. | 0.019        | <b>0.882</b>   | 0.237        | 0.075          | 0.039        | 0.059        | 0.008        | 0.000        | 0.090           | 0.049          | 0.000        | 0.887     |
| Vehicle     | 0.000        | 0.000          | <b>0.001</b> | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | 0.000        | 0.899     |
| Urban Furn. | 0.003        | 0.007          | 0.502        | <b>0.540</b>   | 0.007        | 0.068        | 0.059        | 0.003        | 0.005           | 0.061          | 0.359        | 0.521     |
| Roof        | 0.001        | 0.014          | 0.149        | 0.126          | <b>0.936</b> | 0.046        | 0.003        | 0.002        | 0.000           | 0.004          | 0.427        | 0.944     |
| Facade      | 0.000        | 0.002          | 0.043        | 0.067          | 0.004        | <b>0.726</b> | 0.009        | 0.001        | 0.000           | 0.063          | 0.146        | 0.851     |
| Shrub       | 0.009        | 0.000          | 0.000        | 0.036          | 0.000        | 0.008        | <b>0.512</b> | 0.014        | 0.000           | 0.002          | 0.000        | 0.667     |
| Tree        | 0.004        | 0.000          | 0.002        | 0.026          | 0.001        | 0.012        | 0.306        | <b>0.967</b> | 0.000           | 0.000          | 0.049        | 0.938     |
| Soil/Gravel | 0.069        | 0.008          | 0.001        | 0.001          | 0.000        | 0.000        | 0.000        | 0.000        | <b>0.459</b>    | 0.000          | 0.000        | 0.488     |
| Vert .Surf. | 0.001        | 0.002          | 0.004        | 0.040          | 0.001        | 0.053        | 0.012        | 0.001        | 0.000           | <b>0.774</b>   | 0.018        | 0.539     |
| Chimney     | 0.000        | 0.000          | 0.000        | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.000</b> | 0.000     |
| Recall      | 0.895        | 0.882          | 0.001        | 0.540          | 0.936        | 0.726        | 0.512        | 0.967        | 0.459           | 0.774          | 0.000        | 0.854     |
| F1          | 0.859        | 0.884          | 0.002        | 0.530          | 0.940        | 0.784        | 0.579        | 0.952        | 0.473           | 0.636          | 0.000        | 0.604     |

Table 5.7: Confusion matrix using proposed SCE module on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label.

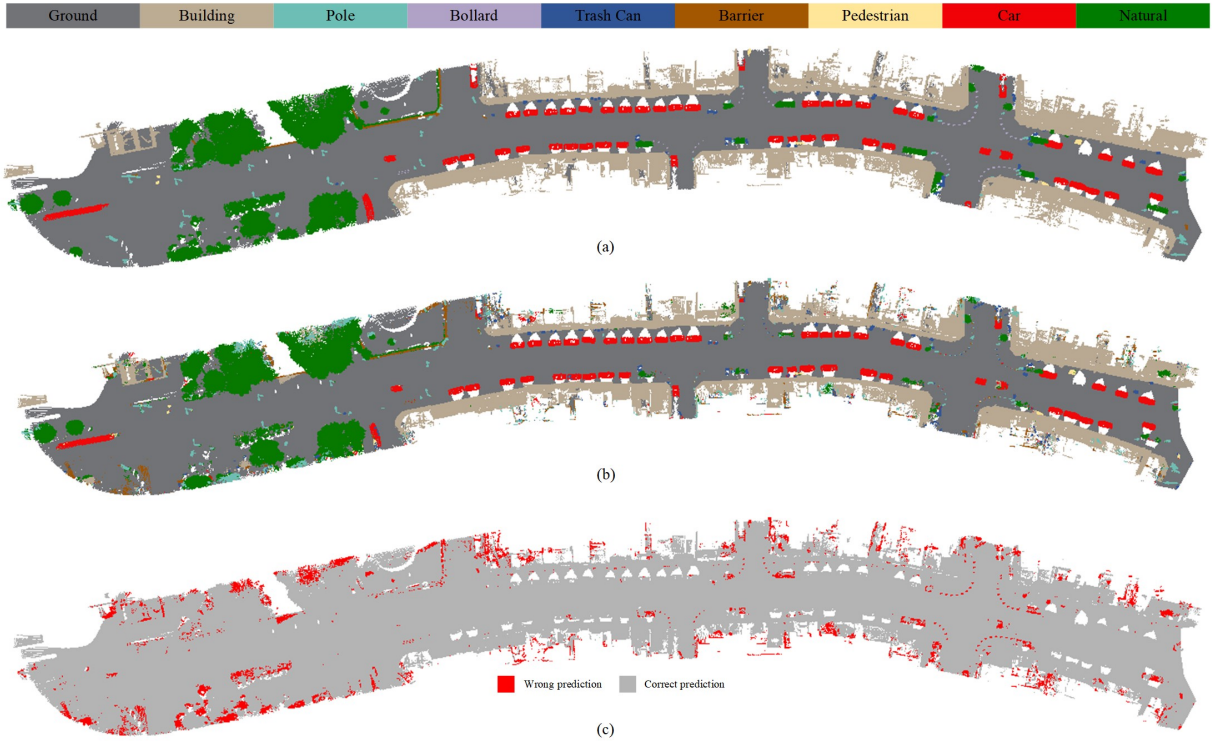


Figure 5.9: Classification results using proposed SCE module on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## Results analysis on the Paris-Lille-3D dataset

To further validate the effectiveness of the proposed SCE module, we conducted both qualitative and quantitative evaluations on the Paris-Lille-3D dataset. As shown in Fig. 5.12, the proposed method achieves segmentation results that are visually superior to the baseline attention-based model in Fig 5.5. Compared with the baseline attention-based network, the SCE module produces more coherent and fine-grained segmentation maps. While maintaining highly accurate segmentation of large-area urban elements such as ground and buildings, the boundaries of fine-grained structures like barriers and natural vegetation patches are rendered with much greater clarity. The error map in Fig. 5.9(c) further highlights the improved segmentation accuracy, with fewer misclassifications and more precise delineation of object boundaries.

| Class      | Ground       | Building     | <u>Pole</u>  | <u>Bollard</u> | <u>Trash Can</u> | Barrier      | <u>Pedestrian</u> | Car          | Natural      | Precision |
|------------|--------------|--------------|--------------|----------------|------------------|--------------|-------------------|--------------|--------------|-----------|
| Ground     | <b>0.997</b> | 0.002        | 0.002        | 0.153          | 0.023            | 0.011        | 0.016             | 0.001        | 0.092        | 0.991     |
| Building   | 0.001        | <b>0.983</b> | 0.024        | 0.000          | 0.083            | 0.085        | 0.015             | 0.002        | 0.041        | 0.990     |
| Pole       | 0.000        | 0.003        | <b>0.877</b> | 0.270          | 0.011            | 0.108        | 0.036             | 0.000        | 0.018        | 0.655     |
| Bollard    | 0.000        | 0.000        | 0.000        | <b>0.000</b>   | 0.000            | 0.000        | 0.000             | 0.000        | 0.000        | 0.000     |
| Trash Can  | 0.000        | 0.002        | 0.002        | 0.193          | <b>0.796</b>     | 0.006        | 0.032             | 0.006        | 0.003        | 0.794     |
| Barrier    | 0.000        | 0.003        | 0.028        | 0.178          | 0.012            | <b>0.632</b> | 0.003             | 0.001        | 0.006        | 0.470     |
| Pedestrian | 0.000        | 0.000        | 0.004        | 0.099          | 0.001            | 0.010        | <b>0.779</b>      | 0.001        | 0.001        | 0.699     |
| Car        | 0.000        | 0.003        | 0.002        | 0.053          | 0.071            | 0.052        | 0.024             | <b>0.988</b> | 0.003        | 0.954     |
| Natural    | 0.002        | 0.004        | 0.062        | 0.053          | 0.003            | 0.095        | 0.094             | 0.001        | <b>0.836</b> | 0.930     |
| Recall     | 0.997        | 0.983        | 0.877        | 0.000          | 0.796            | 0.632        | 0.779             | 0.988        | 0.836        | 0.982     |
| F1         | 0.994        | 0.987        | 0.750        | 0.000          | 0.795            | 0.539        | 0.737             | 0.971        | 0.880        | 0.739     |

Table 5.8: Confusion matrix using proposed SCE module on the Paris-Lille-3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

The quantitative results are summarized in Table 5.8. Compared with the baseline model in Table 5.3, the method using SCE module achieves an increase in mF1 score from 71.5% to 73.9% and improves the OA from 98.0% to 98.2%. The limited gain in OA can be attributed to the dominance of large-area classes such as ground and building, whose accuracy levels are already close to the performance ceiling in the baseline model. In contrast, the mF1 score provides a more balanced evaluation across both dominant and fine-grained categories, and its substantial improvement highlights the enhanced ability of the SCE module to better capture subtle structural details and underrepresented classes. For instance, the F1 score of the trash can class improves from 77.7% to 79.5%, while that of the barrier class increases from 51.2% to 53.9%. For the natural vegetation class with complex boundaries, the F1-score increased from 85.5% to 88.0%. However, for the pole

class, which exhibits less distinctive geometric structures, the F1 score decreased from 83.5% to 75.0%. Such decline may stem from the increased sensitivity of the SCE module to fine boundary details, which occasionally results in more strict classification along ambiguous or indistinct edges. Despite these improvements, challenges remain for highly underrepresented categories such as bollards, which still suffers from the low F1 score of 0.0%. The scarcity of points associated with these rare classes constrains effective feature learning, making it difficult to achieve reliable discrimination.

| Class       | Low<br>Vege. | Impe.<br>Surf. | Vehicle      | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | Chimney      | Precision |
|-------------|--------------|----------------|--------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|--------------|-----------|
| Low Vege.   | <b>0.951</b> | 0.099          | 0.148        | 0.126          | 0.029        | 0.037        | 0.141        | 0.022        | 0.677           | 0.036          | 0.000        | 0.778     |
| Impe. Surf. | 0.020        | <b>0.880</b>   | 0.161        | 0.100          | 0.047        | 0.073        | 0.007        | 0.000        | 0.113           | 0.035          | 0.002        | 0.874     |
| Vehicle     | 0.000        | 0.000          | <b>0.001</b> | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | 0.000        | 1.000     |
| Urban Furn. | 0.001        | 0.004          | 0.486        | <b>0.490</b>   | 0.006        | 0.059        | 0.048        | 0.003        | 0.002           | 0.047          | 0.145        | 0.540     |
| Roof        | 0.000        | 0.008          | 0.148        | 0.129          | <b>0.914</b> | 0.054        | 0.003        | 0.003        | 0.000           | 0.002          | 0.571        | 0.946     |
| Facade      | 0.000        | 0.002          | 0.039        | 0.060          | 0.003        | <b>0.673</b> | 0.004        | 0.001        | 0.001           | 0.101          | 0.198        | 0.849     |
| Shrub       | 0.003        | 0.000          | 0.000        | 0.042          | 0.000        | 0.006        | <b>0.472</b> | 0.011        | 0.000           | 0.001          | 0.000        | 0.731     |
| Tree        | 0.001        | 0.000          | 0.001        | 0.023          | 0.001        | 0.015        | 0.317        | <b>0.959</b> | 0.000           | 0.000          | 0.078        | 0.940     |
| Soil/Gravel | 0.024        | 0.004          | 0.013        | 0.001          | 0.000        | 0.000        | 0.000        | 0.000        | <b>0.207</b>    | 0.004          | 0.000        | 0.534     |
| Vert .Surf. | 0.001        | 0.002          | 0.003        | 0.030          | 0.000        | 0.082        | 0.008        | 0.000        | 0.000           | <b>0.774</b>   | 0.005        | 0.523     |
| Chimney     | 0.000        | 0.000          | 0.000        | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.000</b> | 0.000     |
| Recall      | 0.951        | 0.880          | 0.001        | 0.490          | 0.914        | 0.673        | 0.472        | 0.959        | 0.207           | 0.774          | 0.000        | 0.847     |
| F1          | 0.856        | 0.877          | 0.003        | 0.513          | 0.930        | 0.751        | 0.574        | 0.949        | 0.298           | 0.624          | 0.000        | 0.580     |

Table 5.9: Confusion matrix using early-stage embedding on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label.

## Performance comparison and analysis between early-stage embedding and local context embedding

To assess the representational benefits of the proposed SCE module, we conduct a comparative analysis against the early-stage embedding strategy, which typically performs hard fusion of additional features at the input level. The quantitative evaluation of the early-stage embedding strategy on the Hessigheim 3D dataset are presented in Table 5.9. When comparing the performance metrics with Table 5.7, the model enhanced with local context embedding achieves a higher mF1 score of 60.4% and OA of 85.4%, outperforming the early-stage embedding model which yields a mF1 score of 58.0% and OA of 84.7%. Furthermore, improvements are particularly notable in categories prone to misclassification due to structural similarity, such as a 12.7% gain in the soil/gravel class and a 7.2% gain in vertical facade class. This indicates that while early-stage embedding provides a fast and simple integration of raw features, it is less capable of capturing fine-grained semantic details necessary for complex scene parsing due to their limited capacity for adaptive feature refinement

in contextual or hierarchical settings. The SCE module effectively bridges this gap by enriching intermediate representations with meaningful spatial context, allowing for improved feature discrimination and segmentation accuracy in complex urban environments.

### Performance comparison and analysis between multi-scale and single-scale local context embedding

To further validate the representational strength of the proposed SCE module, we conduct a detailed comparison between its single-scale and multi-scale implementations. The key distinction lies in how local surface features are aggregated: single-scale embedding extracts contextual information from a fixed spatial radius, while multi-scale embedding incorporates features from neighborhoods of varying sizes. Quantitative evaluation on the Hessigheim 3D dataset in Table 5.10 shows that the multi-scale SCE variant consistently outperforms the single-scale configuration in terms of the mF1 score and the OA. It is evident that spatial dependencies extracted from multiple granularity levels of the raw point cloud are crucial for accurately representing classes with diverse scales and structural complexity. Furthermore, categories such as soil/gravel, facades, and vertical surfaces, which are often affected by inter-class confusion due to subtle geometric differences, exhibit noticeable improvements, demonstrating the multi-scale variant can adaptively attend to the most relevant contextual scale. In addition, we also analyze the impact of different neighborhood scales in the single-scale setup to evaluate how the range of contextual reception affects segmentation performance. The experimental results of different scale configurations reveals that including excessively large scales may introduce noise or dilute fine-grained details, while too small scales may lack sufficient context for reliable classification. Therefore, selecting appropriate scale ranges tailored to specific tasks is essential for achieving an optimal balance between local detail and global structural context.

| Scale<br>(cm) | F1 Score     |                |            |                |             |             |             |             |                 |                |         | OA          | mF1         |
|---------------|--------------|----------------|------------|----------------|-------------|-------------|-------------|-------------|-----------------|----------------|---------|-------------|-------------|
|               | Low<br>Vege. | Impe.<br>Surf. | Vehicle    | Urban<br>Furn. | Roof        | Facade      | Shrub       | Tree        | Soil/<br>Gravel | Vert.<br>Surf. | Chimney |             |             |
| –             | 84.6         | 87.5           | 0.0        | 49.7           | 92.8        | 76.4        | 59.0        | 95.2        | 34.6            | 56.4           | 0.0     | 83.8        | 57.8        |
| 5             | 86.2         | <b>88.6</b>    | 0.0        | 51.7           | 93.1        | 76.8        | <b>59.2</b> | 95.1        | 33.9            | 62.9           | 0.0     | 84.9        | 58.9        |
| 10            | <b>86.98</b> | 87.7           | 0.0        | 52.5           | 93.6        | 76.3        | 56.0        | 95.3        | 35.8            | 57.9           | 0.0     | 85.2        | 58.4        |
| 15            | 85.5         | 87.7           | <b>9.5</b> | 52.4           | 93.2        | 76.8        | 58.8        | 95.3        | 32.0            | 57.6           | 0.0     | 84.5        | 59.0        |
| 20            | 85.3         | 87.7           | 0.0        | 52.5           | 93.4        | 76.7        | 54.5        | 95.3        | 16.0            | 59.6           | 0.0     | 84.4        | 56.5        |
| 30            | 86.0         | 87.3           | 0.0        | 47.4           | 93.2        | 75.1        | 58.9        | <b>95.5</b> | 35.6            | 47.7           | 0.0     | 84.5        | 57.0        |
| 10&15         | 85.9         | 88.4           | 0.2        | <b>53.0</b>    | <b>94.0</b> | <b>78.4</b> | 57.9        | 95.2        | <b>47.3</b>     | <b>63.6</b>    | 0.0     | <b>85.4</b> | <b>60.4</b> |

Table 5.10: Comparison of the semantic segmentation results on the Hessigheim 3D dataset using the attention network with different local scales (values in %). Note that the highest values in per-class F1 score, OA, and mF1 are highlighted in bold text.

### Complexity and runtime analysis

To further assess the practicality of the proposed method, we investigate the impact of integrating the proposed SCE module in terms of computational complexity and training efficiency. Specifically, we report and compare the number of parameters, per-iteration training time, and GPU memory usage for models with and without the SCE module on the Hessigheim 3D dataset, as summarized in Table 5.11. The number of parameters serves as an indicator of the overall model complexity, which directly influences computational cost and memory demand. The integration of the SCE module results in a slight increase in the number of parameters, from 11.55M of method A to 11.68M of method B, suggesting a minimal impact on the overall model size. In terms of computational efficiency, method B exhibits a modest increase in training time per iteration, from 511 ms/iter to 538 ms/iter. This is primarily due to the additional operations required by the SCE module to enhance local spatial context representations. Similarly, GPU memory consumption increases from 8.16 GB to 8.29 GB, reflecting the additional memory overhead introduced by the feature aggregation process in the enhancement module. Despite these increases, the added overhead remains moderate and well within acceptable bounds, especially considering the performance gains the SCE module delivers in fine-grained segmentation tasks as demonstrated in Table 5.7. It suggests that the proposed SCE module achieves a favorable balance between model complexity and segmentation performance, offering enhanced representational capacity with only a slight increase in computational resources. Notably, while the increase in model complexity is nearly negligible, extracting original local features during preprocessing remains time-consuming.

| Method | SCE module | Params (millions) | Running time (ms/iter) | Memory (GB) |
|--------|------------|-------------------|------------------------|-------------|
| A      |            | 11.55             | 511                    | 8.16        |
| B      | ✓          | 11.68             | 538                    | 8.29        |

Table 5.11: Comparison of number of parameters, computation time, and GPU memory usage with and without the SCE module. It is estimated and averaged on the Hessigheim 3D dataset.

#### 5.3.3 Contextual joint strategy for rare class augmentation

To address the insufficient representation of rare categories caused by class imbalance in large-scale urban point cloud segmentation, we further introduce a contextual joint augmentation strategy that combines scene-level and class-level enhancements to enrich scene diversity and strengthen rare class representations. Specifically, we evaluate the proposed augmentation strategy for rare

class segmentation using two benchmark datasets, including the Hessigheim 3D dataset and Paris-Lille-3D dataset, and compare the results with the baseline model in Section 5.3.1. Furthermore, the ablation study is conducted to assess the effectiveness of the scene-level and class-level modules in the context joint enhancement strategy.

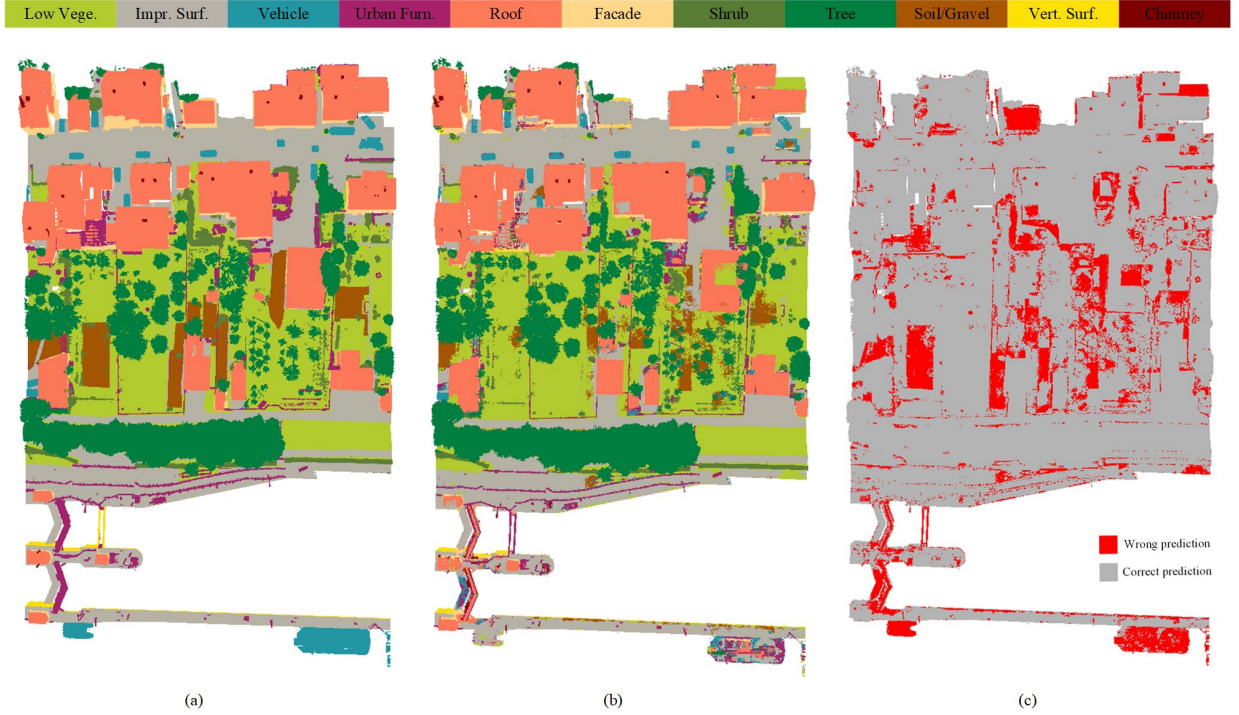


Figure 5.10: Classification results using the contextual joint augmentation on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## Results analysis on the Hessigheim 3D dataset

To thoroughly evaluate the effectiveness of the proposed contextual joint augmentation strategy under complex urban scenarios, both qualitative and quantitative analyses were conducted on the Hessigheim 3D dataset. As shown in the confusion matrix in Table 5.12 and the qualitative segmentation results in Fig. 5.10 and 5.11, the proposed method achieves a notable overall segmentation performance with an OA of 85.4% and a mF1 score of 66.1%. Visualizations of segmentation results in Fig. 5.10 and 5.11 further confirm these findings. With the joint context-aware augmentation applied, the segmentation boundaries for rare categories such as vehicles and chimneys become more distinct, while majority classes exhibit improved spatial consistency.

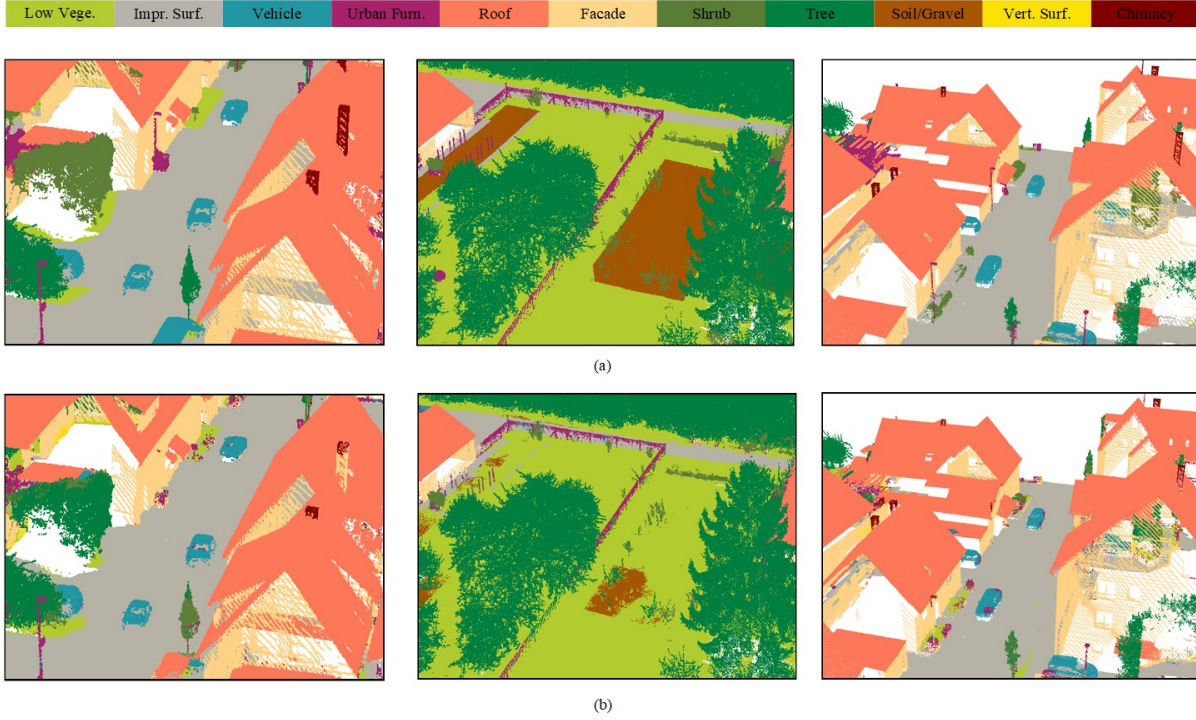
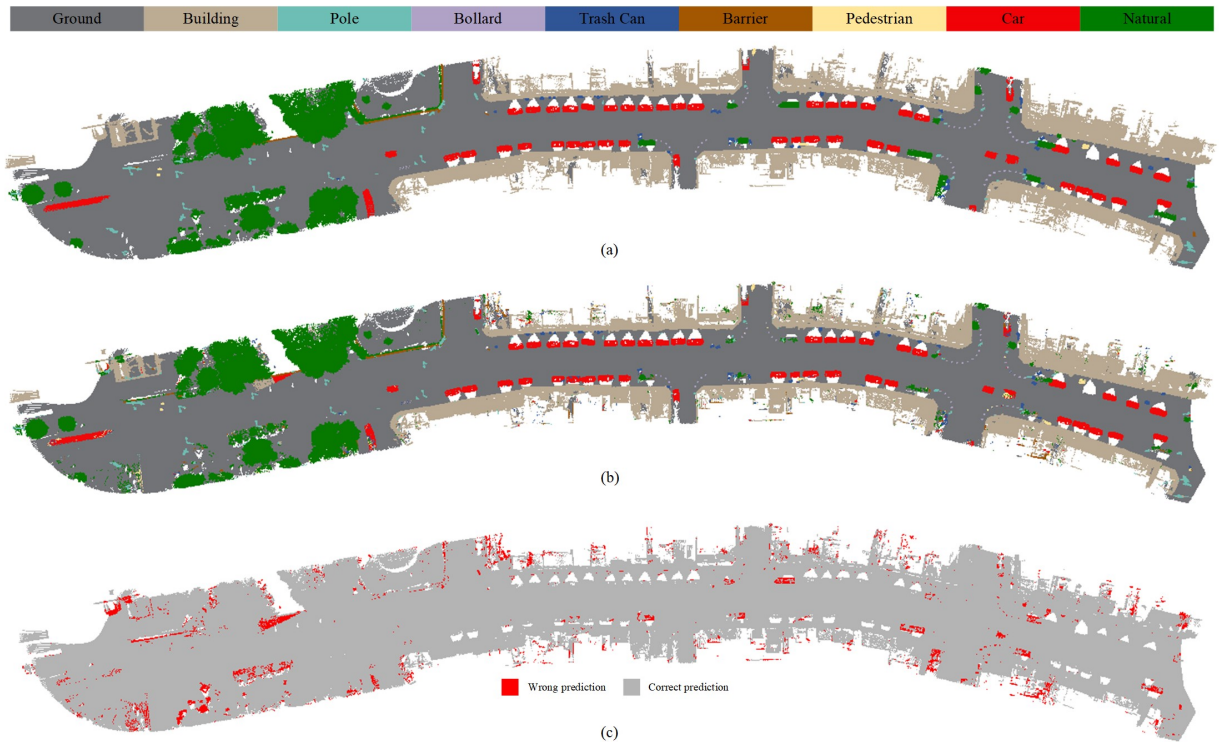


Figure 5.11: Detailed classification results using the contextual joint augmentation on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

| Class       | Low<br>Vege. | Impe.<br>Surf. | <u>Vehicle</u> | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | <u>Chimney</u> | Precision |
|-------------|--------------|----------------|----------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|----------------|-----------|
| Low Vege.   | <b>0.919</b> | 0.071          | 0.026          | 0.100          | 0.023        | 0.024        | 0.116        | 0.014        | 0.605           | 0.006          | 0.000          | 0.813     |
| Impe. Surf. | 0.029        | <b>0.896</b>   | 0.291          | 0.094          | 0.041        | 0.070        | 0.012        | 0.000        | 0.115           | 0.028          | 0.000          | 0.866     |
| Vehicle     | 0.000        | 0.003          | <b>0.384</b>   | 0.049          | 0.003        | 0.008        | 0.005        | 0.001        | 0.001           | 0.006          | 0.000          | 0.571     |
| Urban Furn. | 0.001        | 0.003          | 0.151          | <b>0.434</b>   | 0.005        | 0.043        | 0.067        | 0.002        | 0.002           | 0.062          | 0.057          | 0.632     |
| Roof        | 0.000        | 0.007          | 0.080          | 0.136          | <b>0.923</b> | 0.036        | 0.002        | 0.004        | 0.000           | 0.011          | 0.310          | 0.954     |
| Facade      | 0.001        | 0.005          | 0.049          | 0.088          | 0.004        | <b>0.759</b> | 0.013        | 0.001        | 0.001           | 0.207          | 0.182          | 0.793     |
| Shrub       | 0.007        | 0.000          | 0.000          | 0.035          | 0.000        | 0.007        | <b>0.431</b> | 0.012        | 0.000           | 0.000          | 0.000          | 0.670     |
| Tree        | 0.003        | 0.000          | 0.007          | 0.030          | 0.001        | 0.008        | 0.336        | <b>0.964</b> | 0.000           | 0.001          | 0.004          | 0.936     |
| Soil/Gravel | 0.039        | 0.013          | 0.007          | 0.009          | 0.000        | 0.000        | 0.001        | 0.000        | <b>0.276</b>    | 0.025          | 0.000          | 0.461     |
| Vert. Surf. | 0.001        | 0.002          | 0.002          | 0.015          | 0.000        | 0.042        | 0.017        | 0.001        | 0.000           | <b>0.654</b>   | 0.000          | 0.573     |
| Chimney     | 0.000        | 0.000          | 0.003          | 0.018          | 0.000        | 0.003        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.447</b>   | 0.379     |
| Recall      | 0.919        | 0.896          | 0.384          | 0.434          | 0.923        | 0.759        | 0.431        | 0.964        | 0.276           | 0.654          | 0.447          | 0.854     |
| F1          | 0.863        | 0.881          | 0.459          | 0.515          | 0.939        | 0.776        | 0.525        | 0.950        | 0.345           | 0.611          | 0.410          | 0.661     |

Table 5.12: Confusion matrix using the contextual joint augmentation on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

When compared with the baseline model without augmentation in Table 5.1, the introduction of the contextual joint augmentation results in a significant 8.3% improvement in the mF1 score and an increase of 1.6% in the OA. This performance gain is largely attributed to the enhanced segmentation capability on rare categories. For instance, the F1 score for the vehicle class improves from 0 to 45.9%, and for the chimney class, it increases from 0 to 41.0%. These improvements demonstrate that systematically increasing the number of samples for rare objects can effectively mitigate the data imbalance problem and significantly enhance feature learning for underrepresented classes. Beyond rare categories, the contextual joint augmentation strategy also contributes positively to the segmentation of majority classes and the global structural understanding of urban scenes. For example, the F1 scores for the facade and low vegetation classes increase by 4.7% and 1.7%, respectively. It suggests that sub-scene masking mechanism not only augments scene diversity but also enhances the semantic interaction in global structures, thereby improving the robustness and adaptability of the model in large-scale and complex urban environments.



*Figure 5.12:* Classification results using the contextual joint augmentation on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## Results analysis on the Paris-Lille-3D dataset

To conduct a comprehensive evaluation, we further validated the effectiveness of the proposed contextual joint augmentation strategy on the MLS-based dataset Paris-Lille-3D, which features diverse object-level variations in urban street environments. The visualization results in Fig. 5.12 show that the segmentation results using contextual joint augmentation exhibit more precise and coherent boundaries for various object categories, especially for rare and small-scale classes such as bollards, trash cans, barriers, and pedestrians. The error map in Fig. 5.12(c) further reveals the reduction of misclassifications in rare object regions and along class boundaries, which demonstrates enhanced structural awareness and semantic consistency.

| Class      | Ground       | Building     | <u>Pole</u>  | <u>Bollard</u> | <u>Trash Can</u> | Barrier      | <u>Pedestrian</u> | Car          | Natural      | Precision |
|------------|--------------|--------------|--------------|----------------|------------------|--------------|-------------------|--------------|--------------|-----------|
| Ground     | <b>0.996</b> | 0.004        | 0.004        | 0.182          | 0.045            | 0.017        | 0.033             | 0.002        | 0.108        | 0.989     |
| Building   | 0.001        | <b>0.984</b> | 0.031        | 0.007          | 0.057            | 0.027        | 0.039             | 0.000        | 0.040        | 0.991     |
| Pole       | 0.000        | 0.003        | <b>0.899</b> | 0.088          | 0.005            | 0.104        | 0.014             | 0.000        | 0.008        | 0.728     |
| Bollard    | 0.000        | 0.000        | 0.001        | <b>0.614</b>   | 0.000            | 0.015        | 0.013             | 0.000        | 0.000        | 0.708     |
| Trash Can  | 0.000        | 0.001        | 0.000        | 0.037          | <b>0.826</b>     | 0.009        | 0.004             | 0.004        | 0.002        | 0.873     |
| Barrier    | 0.001        | 0.003        | 0.020        | 0.003          | 0.005            | <b>0.686</b> | 0.000             | 0.001        | 0.008        | 0.511     |
| Pedestrian | 0.000        | 0.001        | 0.002        | 0.019          | 0.001            | 0.014        | <b>0.879</b>      | 0.003        | 0.005        | 0.422     |
| Car        | 0.000        | 0.002        | 0.001        | 0.020          | 0.062            | 0.022        | 0.003             | <b>0.990</b> | 0.000        | 0.967     |
| Natural    | 0.001        | 0.003        | 0.043        | 0.030          | 0.000            | 0.107        | 0.014             | 0.001        | <b>0.830</b> | 0.945     |
| Recall     | 0.996        | 0.984        | 0.899        | 0.614          | 0.826            | 0.686        | 0.879             | 0.990        | 0.830        | 0.982     |
| F1         | 0.993        | 0.988        | 0.804        | 0.657          | 0.849            | 0.586        | 0.570             | 0.979        | 0.883        | 0.812     |

Table 5.13: Confusion matrix using the contextual joint augmentation on the Paris-Lille-3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

Compared with the baseline results in Table 5.3, the proposed method in Table 5.13 achieves a notable improvement in overall segmentation performance, with an increase in the mF1 score from 71.5% to 81.2% and a stable OA of 98.2%. It demonstrates that the proposed augmentation strategy not only stabilizes the overall segmentation accuracy but also ensures a balanced capability for both rare and majority classes in the entire scene. Although the improvement in the OA is moderate, with an increase from 98.0% to 98.2%, the most remarkable gains are observed in rare and structurally complex categories. For instance, the F1 score of the trash can class increases from 77.7% to 84.9%, and for barriers, it rises from 51.2% to 58.6%, while bollards improve significantly from 0 to 65.7%. This strongly indicates that the contextual joint augmentation effectively alleviates the data imbalance problem by providing additional learning instances and contextually consistent sub-scene augmentations for underrepresented classes. However, a slight decrease is observed in the

F1 score for the pole class, dropping from 83.5% to 80.4%. This decline may be attributed to the increased confusion with structurally similar objects, such as bollards and barriers, especially as the augmentation process introduces more instances of rare and compact classes. Such added diversity in small-scale object representations may have unintentionally introduced ambiguity around slender vertical structures, leading to a higher rate of misclassification.

## Ablation study

To further investigate the individual contributions of the scene-level and class-level strategies within the proposed contextual joint augmentation, we conducted a series of ablation experiments on the Hessigheim 3D dataset. These experiments isolate the effects of sub-scene masking modules in the scene-level strategy and dual augmentation modules in the class-level strategy, respectively. The comparative results of these experiments are detailed in Table 5.14. Compared with the baseline method A, both the scene-level augmentation and class-level augmentation strategies individually contribute to improved segmentation performance in complex urban environments. Specifically, method B, which applies only scene-level augmentation, yields a moderate enhancement in overall segmentation, with an increase of 0.7% in the OA and 1.2% in the mF1 score over the baseline method. The improvement is primarily attributed to the sub-scene masking module, which promotes semantic interactions between majority classes.

| Method | Augmentation |             | F1 Score    |             |                |             |             |             |             |             |             |             |                | OA          | mF1         |
|--------|--------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|-------------|-------------|
|        | Scene-level  | Class-level | Low Vege.   | Impe. Surf. | <u>Vehicle</u> | Urban Furn. | Roof        | Facade      | Shrub       | Tree        | Soil/Gravel | Vert. Surf. | <u>Chimney</u> |             |             |
| A      |              |             | 84.6        | 87.5        | 0.0            | 49.7        | 92.8        | 76.4        | 59.0        | 95.2        | 34.6        | 56.4        | 0.0            | 83.8        | 57.8        |
| B      | ✓            |             | 85.6        | 86.4        | 0.0            | 51.4        | 93.5        | 74.6        | 59.1        | 95.1        | <b>45.2</b> | 58.4        | 0.0            | 84.5        | 59.0        |
| C      |              | ✓           | 84.6        | 87.1        | 43.7           | <b>56.0</b> | <b>93.9</b> | <b>77.7</b> | <b>59.3</b> | <b>95.2</b> | 29.4        | 46.9        | 14.5           | 84.5        | 62.6        |
| D      | ✓            | ✓           | <b>86.3</b> | <b>88.1</b> | <b>45.9</b>    | 51.5        | 93.9        | 77.6        | 52.5        | 95.0        | 34.5        | <b>61.1</b> | <b>41.0</b>    | <b>85.4</b> | <b>66.1</b> |

Table 5.14: Comparison of the semantic segmentation results using the Hessigheim 3D dataset with different augmentation strategies (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, and mF1 are highlighted in bold text.

However, the lack of targeted augmentation for underrepresented categories limits its effectiveness in rare class recognition. In contrast, method C, which employs only class-level augmentation, demonstrates a substantial gains in the segmentation performance of rare objects. For example, the F1 scores of the vehicle and the chimney categories improve by 43.7% and 14.4% compared to the baseline method, respectively. Such tailored augmentation leads to a significant 4.7% increase in the mF1 score, demonstrating the effectiveness of this strategy in alleviating the effects of class imbalance and enhancing the discriminative feature learning for rare classes. Notably, the best performance is achieved by method D, which combines both augmentation strategies in a unified

framework, with an OA of 85.4% and a mF1 score of 66.1%. In this setting, the F1 score of the rare classes including vehicles and chimneys reach 45.9% and 41.0%, respectively. It indicates that the scene-level augmentation improves the model’s ability to capture global contextual dependencies, while the class-level augmentation compensates for data sparsity in rare classes. Their complementary integration results in simultaneous improvements in global semantic consistency and rare class segmentation accuracy. Nevertheless, the F1 score of the shrub class declines by 6.5%. A potential explanation is that the addition of diverse augmentation strategies may introduce unrealistic spatial distributions for certain classes, leading to confusion in their representation.

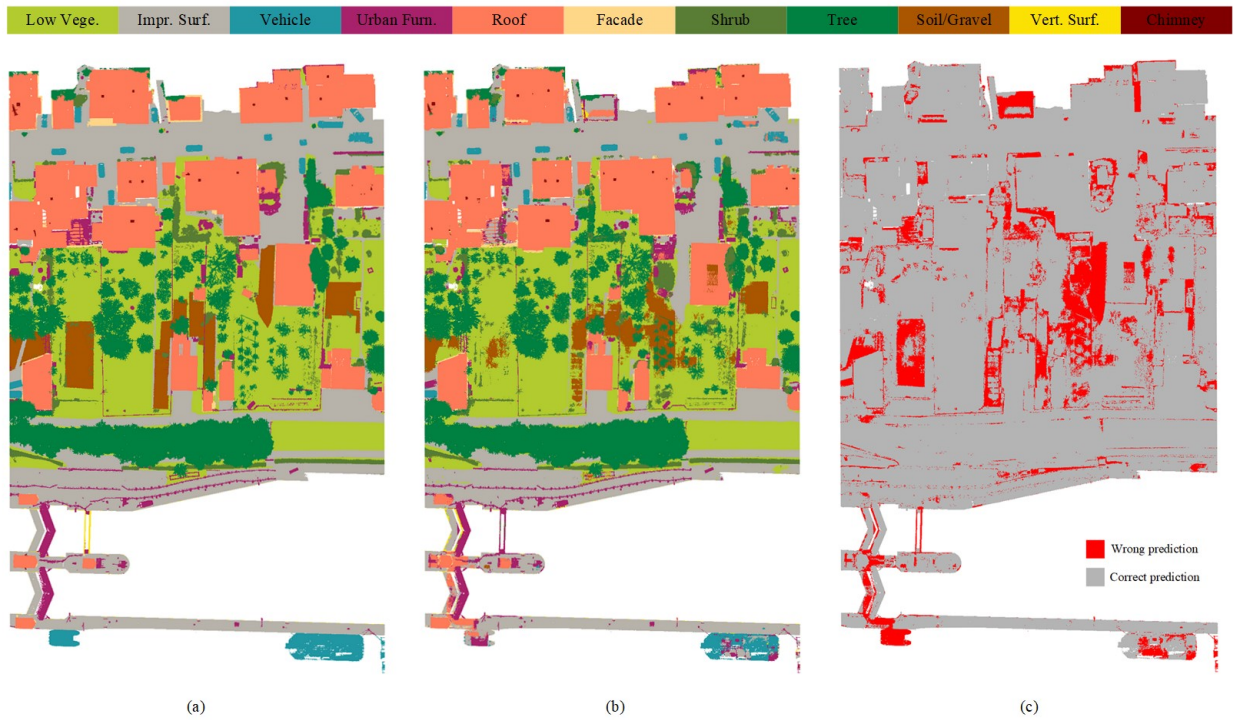
In conclusion, the proposed contextual joint augmentation strategy achieves a balanced improvement in both rare class segmentation and overall scene understanding in large-scale urban point cloud tasks. By integrating multi-level augmentation mechanisms, the method systematically enhances the structural diversity of sub-scenes and strengthens the feature representation of rare classes, effectively alleviating the class imbalance issue in complex real-world environments. It is important to note that the current framework primarily improves performance through data-level interventions, such as increasing scene diversity and the number of rare class samples, but does not yet fully exploit the intrinsic geometric and spatial characteristics of rare instances at the individual level. Future work may focus on explicit spatial shape cues of rare targets, through the incorporation of adaptive feature modulation mechanisms, to further enhance segmentation performance of geometrically distinctive rare classes.

#### 5.3.4 Target-aware attention for rare class segmentation

This section presents a comprehensive evaluation of the proposed target-aware attention method for rare class segmentation on four benchmark datasets, including the Hessigheim 3D dataset, the Paris-Lille-3D dataset, the Semantic3D dataset, and the WHU-Urban3D dataset. We compare our TaaNet with several representative deep learning-based approaches, namely PointNet++ (Qi et al., 2017b), MinkowskiNet (Choy et al., 2019), DGCNN (Wang et al., 2019b), PAConv (Xu et al., 2021b), and SPVConv (Tang et al., 2020), which serve as strong baselines in point cloud segmentation tasks. For a fair comparison, the training and test data are organized as specified in Section 4.1. All methods are trained with the same sub-region size, overlap, and number of sampled points following Section 5.2. The generalization capability of the model is further examined on the MLS-W subset of the WHU-Urban3D dataset. Additionally, the contribution of the proposed target-aware aggregator is investigated through an ablation study conducted on the Hessigheim 3D dataset. The section concludes with a detailed analysis of the model complexity.

## Results on the Hessigheim 3D dataset

We initially present the qualitative results of the proposed TaaNet model. The visualized classification outcomes and the corresponding error distribution are provided in Fig. 5.13, while more granular and localized classification results are depicted in Fig. 5.14. As demonstrated in Fig. 5.13(b), the proposed TaaNet achieves impressive performance in the segmentation task, accurately identifying the majority of 3D points in the test scenes. To facilitate a more in-depth understanding of the efficacy of our method, we showcase several representative local regions in Fig. 5.14. These illustrations further validate strong capability of our TaaNet in recognizing challenging rare classes including vehicles and chimneys. The corresponding error visualization in Fig. 5.13(c) illustrates the spatial distribution of prediction errors on the Hessigheim 3D dataset, revealing that the majority of points are correctly classified. However, notable classification errors remain and are generally unevenly distributed in the top-down perspective. When compared with the ground truth labels shown in Fig. 5.13(a), it becomes evident that misclassifications are predominantly concentrated along complex or ambiguous class boundaries, especially between low vegetation and soil/gravel, as well as within fine-grained vegetation subcategories such as shrubs and trees. Such errors are largely due to the high degree of intra-class similarity in structural features between these classes.



*Figure 5.13:* Classification results of our TaaNet model on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

The quantitative evaluation results of our model are reported through the classification confusion matrix in Table 5.15, which includes standard evaluation metrics such as precision, recall, per-class F1 scores, OA, and mF1. Our model achieves an OA of 88.2% and a mF1 of 79.0%, reflecting its robust performance for the majority of semantic classes. Notably, the model excels at classifying rare object classes, achieving F1 scores of 81.5% and 85.5% for vehicles and chimneys, respectively. In contrast, the lowest performance is observed in the soil/gravel class, which is frequently confused with low vegetation as shown by the confusion matrix. This can be attributed to their highly overlapping spatial distribution and the geometric similarity, especially when vegetation is sparse or at its early stages of growth. Additionally, points belonging to shrubs are often misclassified as trees, which can be attributed to their intrinsic structural similarities and the lack of distinct discriminative features at the local scale. Accurate distinction between these classes is compromised by the minimal inter-class distances observed in adjacent local features.

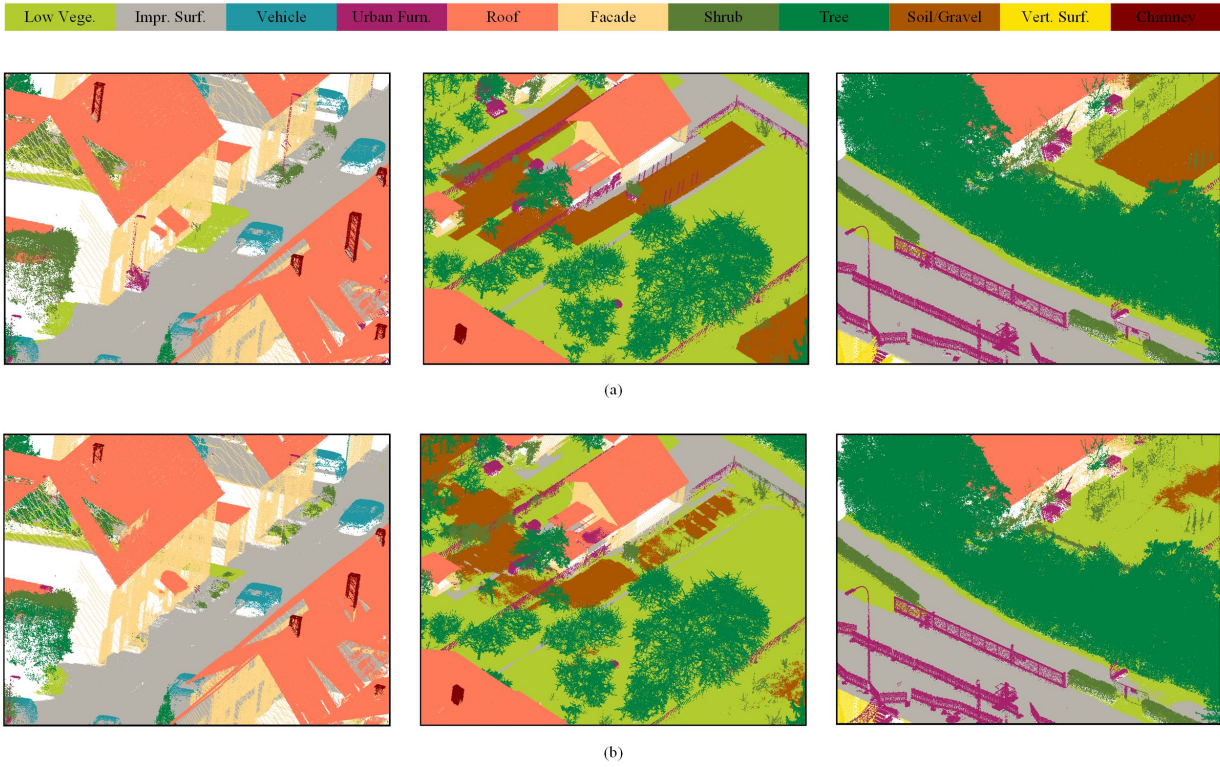


Figure 5.14: Detailed classification results of our TaaNet model on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

To conduct a thorough evaluation, our TaaNet is compared with baseline methods described in Section 5.3.4. As detailed in Table 5.16, TaaNet attains the highest overall accuracy of 88.2%,

outperforming all other referenced methods. Moreover, it yields a notable mF1 score of 79.0%, representing significant relative improvements of 15.2%, 6.4%, 4.5%, 2.0%, and 5.4% over the respective competing models. To further assess performance on imbalanced categories, we introduce the rare-class mean F1 score (rmF1) as an additional evaluation metric. TaaNet achieves an rmF1 of 83.5%, surpassing the second-best method by a considerable margin of 9.9%. In particular, the model obtains impressive F1 scores of 81.5% and 85.5% for the vehicle and chimney classes, respectively. It demonstrates the effectiveness of the proposed TaaNet in enhancing segmentation performance for both majority and rare classes.

| Class       | Low<br>Vege. | Impe.<br>Surf. | <u>Vehicle</u> | Urban<br>Furn. | Roof         | Facade       | Shrub        | Tree         | Soil/<br>Gravel | Vert.<br>Surf. | <u>Chimney</u> | Precision |
|-------------|--------------|----------------|----------------|----------------|--------------|--------------|--------------|--------------|-----------------|----------------|----------------|-----------|
| Low Vege.   | <b>0.887</b> | 0.053          | 0.001          | 0.043          | 0.000        | 0.011        | 0.039        | 0.005        | 0.441           | 0.000          | 0.000          | 0.873     |
| Impe. Surf. | 0.029        | <b>0.915</b>   | 0.131          | 0.042          | 0.023        | 0.036        | 0.012        | 0.000        | 0.144           | 0.012          | 0.000          | 0.899     |
| Vehicle     | 0.000        | 0.000          | <b>0.703</b>   | 0.004          | 0.001        | 0.001        | 0.001        | 0.000        | 0.000           | 0.001          | 0.000          | 0.969     |
| Urban Furn. | 0.003        | 0.008          | 0.117          | <b>0.657</b>   | 0.015        | 0.045        | 0.027        | 0.001        | 0.004           | 0.068          | 0.033          | 0.668     |
| Roof        | 0.000        | 0.011          | 0.031          | 0.050          | <b>0.949</b> | 0.030        | 0.003        | 0.001        | 0.000           | 0.001          | 0.123          | 0.972     |
| Facade      | 0.002        | 0.007          | 0.010          | 0.100          | 0.007        | <b>0.840</b> | 0.017        | 0.001        | 0.002           | 0.106          | 0.026          | 0.798     |
| Shrub       | 0.020        | 0.001          | 0.000          | 0.054          | 0.000        | 0.008        | <b>0.655</b> | 0.021        | 0.004           | 0.015          | 0.000          | 0.584     |
| Tree        | 0.005        | 0.000          | 0.000          | 0.041          | 0.002        | 0.005        | 0.247        | <b>0.970</b> | 0.000           | 0.000          | 0.011          | 0.943     |
| Soil/Gravel | 0.053        | 0.005          | 0.000          | 0.001          | 0.004        | 0.000        | 0.000        | 0.000        | <b>0.405</b>    | 0.000          | 0.000          | 0.516     |
| Vert. Surf. | 0.000        | 0.002          | 0.007          | 0.009          | 0.001        | 0.024        | 0.001        | 0.000        | 0.000           | <b>0.796</b>   | 0.000          | 0.741     |
| Chimney     | 0.000        | 0.000          | 0.000          | 0.000          | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000          | <b>0.808</b>   | 0.908     |
| Recall      | 0.887        | 0.915          | 0.703          | 0.657          | 0.949        | 0.840        | 0.655        | 0.970        | 0.405           | 0.796          | 0.808          | 0.882     |
| F1          | 0.880        | 0.906          | 0.815          | 0.662          | 0.960        | 0.818        | 0.618        | 0.956        | 0.454           | 0.767          | 0.855          | 0.790     |

Table 5.15: Confusion matrix of our TaaNet on the Hessigheim 3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

| Method        | F1 Score     |                |                |                |             |             |             |             |                 |                |                | OA          | mF1         | rmF1        |
|---------------|--------------|----------------|----------------|----------------|-------------|-------------|-------------|-------------|-----------------|----------------|----------------|-------------|-------------|-------------|
|               | Low<br>Vege. | Impe.<br>Surf. | <u>Vehicle</u> | Urban<br>Furn. | Roof        | Facade      | Shrub       | Tree        | Soil/<br>Gravel | Vert.<br>Surf. | <u>Chimney</u> |             |             |             |
| PointNet++    | 88.3         | 88.9           | 49.0           | 43.8           | 92.3        | 71.5        | 49.8        | 94.8        | <b>55.5</b>     | 68.4           | 0.0            | 85.8        | 63.9        | 24.5        |
| MinkowskiNet  | 87.9         | 89.5           | 53.7           | 58.2           | 94.6        | 77.0        | 54.9        | 94.3        | 38.3            | 71.7           | 79.0           | 86.6        | 72.6        | 66.4        |
| DGCNN         | 85.4         | 89.7           | 80.7           | 56.6           | 94.9        | 78.1        | <b>63.0</b> | <b>96.2</b> | 48.1            | 76.1           | 51.0           | 86.6        | 74.5        | 65.8        |
| PACConv       | <b>88.4</b>  | 90.5           | 73.6           | 63.0           | 95.8        | 81.4        | 58.2        | 95.7        | 45.9            | <b>81.7</b>    | 73.6           | 88.1        | 77.1        | 73.6        |
| SPVConv       | 88.2         | 88.6           | 54.2           | 61.9           | 94.8        | 77.7        | 59.2        | 95.0        | 43.0            | 65.4           | 82.2           | 86.9        | 73.7        | 68.2        |
| TaaNet (Ours) | 88.0         | <b>90.6</b>    | <b>81.5</b>    | <b>66.2</b>    | <b>96.0</b> | <b>81.8</b> | 61.8        | 95.6        | 45.4            | 76.7           | <b>85.5</b>    | <b>88.2</b> | <b>79.0</b> | <b>83.5</b> |

Table 5.16: Comparison of the semantic segmentation results on the Hessigheim 3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

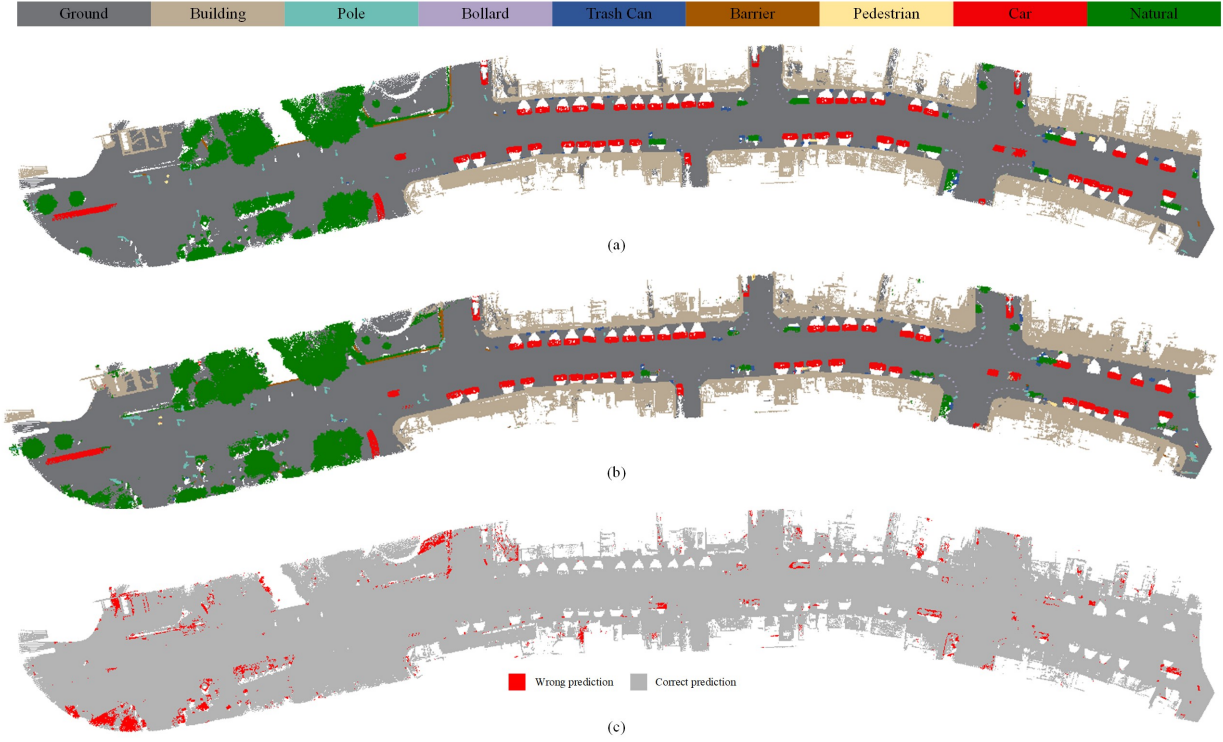


Figure 5.15: Classification results of our TaaNet model on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## Results on the Paris-Lille-3D dataset

In addition to the evaluations conducted on the ALS-based Hessigheim 3D dataset, we further extend our experimental investigation to the MLS-based Paris-Lille-3D dataset, aiming at examining the adaptability and robustness of the proposed TaaNet framework under varying data acquisition modalities and scene complexities. Adopting the same evaluation strategy as employed previously, the qualitative point-wise classification results are visualized in Fig. 5.15, while the corresponding quantitative metrics are reported in Table 5.17 and Table 5.18. As can be observed from Fig. 5.15, the majority of 3D points are accurately predicted, with clear and stable segmentation boundaries maintained between distinct objects. It indicates the capacity of the TaaNet to effectively distinguish between semantically and structurally diverse object types within dense urban environments. Furthermore, the classification confusion matrix provided in Table 5.17 reveals that our model performs particularly well in recognizing rare yet semantically important objects such as poles, bollards, and trash cans. Nevertheless, misclassifications are frequently observed between pedestrians and buildings. This issue is primarily attributed to the spatial proximity between the

two categories: pedestrian points are often densely distributed adjacent to building facades, resulting in overlapping geometric features that hinder effective discrimination. Additionally, since the MLS system captures dynamic scenes, moving pedestrians are not always represented as discrete entities. The clustering process used for generating supervisory signals tends to aggregate multiple pedestrians into a single unit, which limits the ability of the model to learn discriminative features for individual pedestrian instances.

| Class      | Ground       | Building     | <u>Pole</u>  | <u>Bollard</u> | <u>Trash Can</u> | Barrier      | <u>Pedestrian</u> | Car          | Natural      | Precision |
|------------|--------------|--------------|--------------|----------------|------------------|--------------|-------------------|--------------|--------------|-----------|
| Ground     | <b>0.995</b> | 0.002        | 0.002        | 0.111          | 0.025            | 0.007        | 0.018             | 0.000        | 0.053        | 0.995     |
| Building   | 0.002        | <b>0.997</b> | 0.033        | 0.001          | 0.078            | 0.011        | 0.314             | 0.003        | 0.021        | 0.992     |
| Pole       | 0.000        | 0.000        | <b>0.918</b> | 0.006          | 0.000            | 0.078        | 0.000             | 0.000        | 0.004        | 0.917     |
| Bollard    | 0.000        | 0.000        | 0.000        | <b>0.842</b>   | 0.000            | 0.008        | 0.000             | 0.000        | 0.000        | 0.921     |
| Trash Can  | 0.000        | 0.000        | 0.000        | 0.001          | <b>0.861</b>     | 0.004        | 0.000             | 0.000        | 0.003        | 0.966     |
| Barrier    | 0.000        | 0.000        | 0.003        | 0.026          | 0.005            | <b>0.721</b> | 0.003             | 0.000        | 0.005        | 0.803     |
| Pedestrian | 0.000        | 0.000        | 0.000        | 0.000          | 0.000            | 0.010        | <b>0.659</b>      | 0.000        | 0.000        | 0.929     |
| Car        | 0.000        | 0.001        | 0.000        | 0.011          | 0.005            | 0.017        | 0.001             | <b>0.992</b> | 0.000        | 0.987     |
| Natural    | 0.003        | 0.001        | 0.044        | 0.001          | 0.026            | 0.144        | 0.004             | 0.005        | <b>0.915</b> | 0.929     |
| Recall     | 0.995        | 0.997        | 0.918        | 0.842          | 0.861            | 0.721        | 0.659             | 0.992        | 0.915        | 0.990     |
| F1         | 0.995        | 0.994        | 0.918        | 0.880          | 0.910            | 0.760        | 0.771             | 0.989        | 0.922        | 0.904     |

Table 5.17: Confusion matrix of our TaaNet on the Paris-Lille-3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

| Method        | F1 Score    |             |             |                |                  |             |                   |             |             | OA          | mF1         | rmF1        |
|---------------|-------------|-------------|-------------|----------------|------------------|-------------|-------------------|-------------|-------------|-------------|-------------|-------------|
|               | Ground      | Building    | <u>Pole</u> | <u>Bollard</u> | <u>Trash Can</u> | Barrier     | <u>Pedestrian</u> | Car         | Natural     |             |             |             |
| PointNet++    | 99.2        | 96.4        | 63.1        | 0.0            | 21.8             | 12.0        | 7.5               | 94.2        | 75.5        | 95.9        | 52.2        | 23.1        |
| MinkowskiNet  | 99.2        | 99.0        | 83.1        | 84.0           | 84.5             | 52.9        | <b>78.7</b>       | 98.5        | 88.7        | 98.4        | 85.4        | 82.6        |
| DGCNN         | 99.4        | 99.3        | 80.7        | 66.4           | 69.5             | 51.6        | 75.6              | 98.3        | 86.7        | 98.4        | 80.8        | 73.0        |
| PACConv       | 99.4        | 99.0        | 85.1        | 65.9           | 80.6             | 52.1        | 77.1              | 97.2        | 90.4        | 98.5        | 83.0        | 77.2        |
| SPVConv       | 99.3        | 99.2        | 86.9        | 87.5           | 89.1             | 46.3        | 77.6              | 97.6        | 88.0        | 98.5        | 85.7        | 85.3        |
| TaaNet (Ours) | <b>99.5</b> | <b>99.4</b> | <b>91.8</b> | <b>88.0</b>    | <b>91.0</b>      | <b>76.0</b> | 77.1              | <b>98.9</b> | <b>92.2</b> | <b>99.0</b> | <b>90.4</b> | <b>87.0</b> |

Table 5.18: Comparison of the semantic segmentation results on the Paris-Lille-3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

To comprehensively assess performance, we compare TaaNet with all baseline methods previously discussed in Section 5.3.4, with the comparative results summarized in Table 5.18. Our method achieves the best results among all compared approaches, with improvements of 4.7% and

0.5% in the mF1 score and the OA, respectively. While the increase in the OA is relatively modest, this can be attributed to the extremely low frequency of rare classes in the dataset, which limits their influence on the OA metric. Importantly, for rare semantic categories such as poles, bollards, and trash cans, our TaaNet consistently achieves the best performance between all baseline methods, highlighting its robustness and effectiveness in handling class imbalance within complex urban scenes.

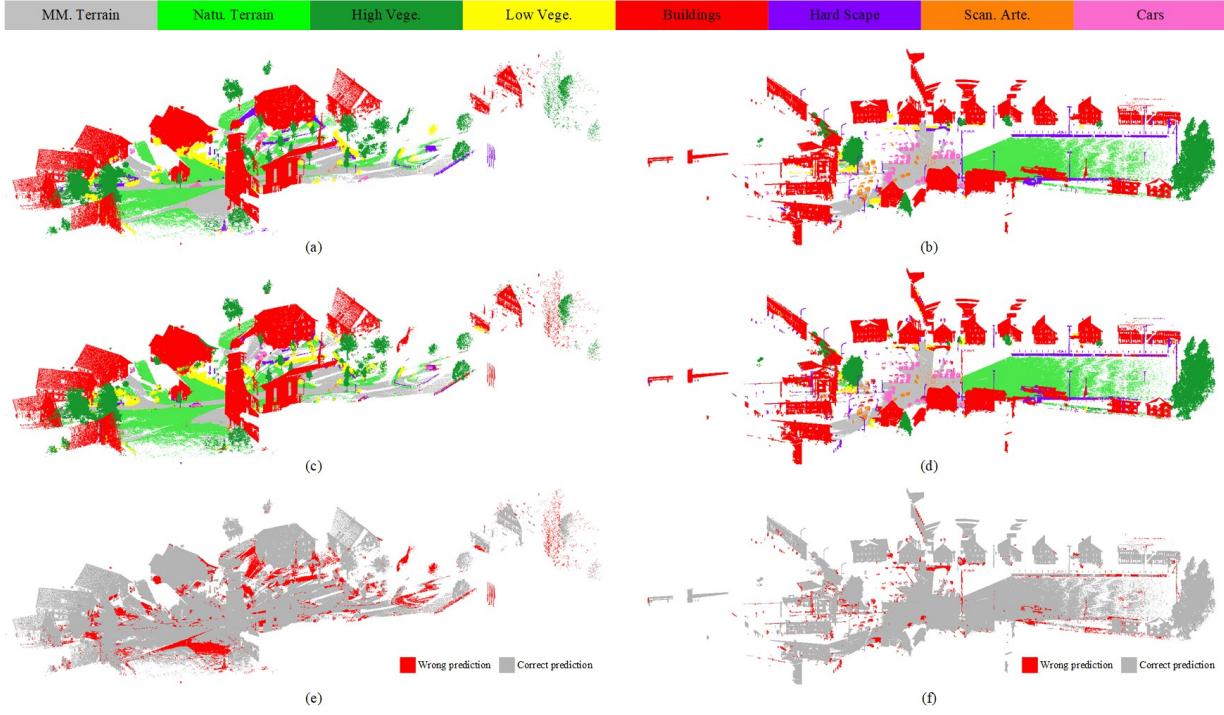


Figure 5.16: Classification results of our TaaNet model on the Semantic3D dataset. Only two selected scenes (bildstein5 and sg27\_9) are shown. (a) and (b) represent the ground truth. (c) and (d) present the classification results using the proposed method. (e) and (f) present the corresponding error maps.

## Results on the Semantic3D dataset

To enable a more comprehensive assessment, we further validate the effectiveness of the proposed TaaNet on the TLS-based Semantic3D dataset. The qualitative results generated by our approach are presented in Fig. 5.16, which demonstrate that the majority of 3D points with different semantic categories are accurately classified. In addition, the quantitative classification outcomes are summarized as the confusion matrix in Table 5.19. It is notable that the primary source of classification confusion occurs between the hard scapes and building categories. A reasonable explanation is that a significant number of points within these two classes share highly similar local structures, which results in ambiguous geometric features. Furthermore, both classes are often constructed using sim-

ilar materials, leading to comparable reflectance and intensity patterns in the LiDAR data, which further complicates the accurate semantic separation.

| Class         | MM.<br>Terrain | Natu.<br>Terrain | High<br>Vege. | Low<br>Vege. | Buildings    | Hard<br>Scape | Scan.<br>Arte. | <u>Cars</u>  | Precision |
|---------------|----------------|------------------|---------------|--------------|--------------|---------------|----------------|--------------|-----------|
| MM. Terrain   | <b>0.979</b>   | 0.014            | 0.000         | 0.000        | 0.003        | 0.006         | 0.019          | 0.002        | 0.989     |
| Natu. Terrain | 0.012          | <b>0.963</b>     | 0.001         | 0.009        | 0.001        | 0.015         | 0.000          | 0.003        | 0.927     |
| High Vege.    | 0.000          | 0.003            | <b>0.962</b>  | 0.066        | 0.002        | 0.008         | 0.002          | 0.001        | 0.944     |
| Low Vege.     | 0.001          | 0.008            | 0.027         | <b>0.751</b> | 0.005        | 0.034         | 0.002          | 0.001        | 0.719     |
| Buildings     | 0.003          | 0.011            | 0.008         | 0.145        | <b>0.968</b> | 0.430         | 0.037          | 0.042        | 0.952     |
| Hard Scape    | 0.002          | 0.002            | 0.002         | 0.029        | 0.020        | <b>0.415</b>  | 0.021          | 0.077        | 0.563     |
| Scan. Arte.   | 0.000          | 0.000            | 0.000         | 0.000        | 0.002        | 0.083         | <b>0.904</b>   | 0.017        | 0.654     |
| Cars          | 0.003          | 0.000            | 0.000         | 0.001        | 0.000        | 0.009         | 0.015          | <b>0.857</b> | 0.949     |
| Recall        | 0.979          | 0.963            | 0.962         | 0.751        | 0.968        | 0.415         | 0.904          | 0.857        | 0.937     |
| F1            | 0.984          | 0.945            | 0.953         | 0.735        | 0.960        | 0.478         | 0.759          | 0.901        | 0.839     |

Table 5.19: Confusion matrix of our TaaNet on the Semantic3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

| Method        | F1 Score       |                  |               |              |             |               |                |             | OA          | mF1         | rmF1        |
|---------------|----------------|------------------|---------------|--------------|-------------|---------------|----------------|-------------|-------------|-------------|-------------|
|               | MM.<br>Terrain | Natu.<br>Terrain | High<br>Vege. | Low<br>Vege. | Buildings   | Hard<br>Scape | Scan.<br>Arte. | <u>Cars</u> |             |             |             |
| PointNet++    | 96.2           | <b>96.4</b>      | 82.1          | 56.8         | 94.4        | 34.9          | 47.3           | 75.3        | 90.5        | 72.9        | 75.3        |
| MinkowskiNet  | 95.6           | 78.3             | 86.0          | 65.2         | 93.3        | 42.4          | 71.9           | 80.4        | 88.9        | 76.6        | 80.4        |
| DGCNN         | 95.6           | 85.0             | 91.0          | 61.6         | 94.9        | 19.0          | 55.4           | 83.4        | 90.3        | 73.2        | 83.4        |
| PACConv       | 97.2           | 88.3             | 91.5          | 50.4         | 95.4        | 45.6          | 54.3           | 85.2        | 91.2        | 76.0        | 85.2        |
| SPVConv       | 96.1           | 78.1             | 82.3          | 65.5         | 93.4        | <b>48.0</b>   | 72.3           | 82.2        | 89.7        | 77.2        | 82.2        |
| TaaNet (Ours) | <b>98.4</b>    | 94.5             | <b>95.3</b>   | <b>73.5</b>  | <b>96.0</b> | 47.8          | <b>75.9</b>    | <b>90.1</b> | <b>93.7</b> | <b>83.9</b> | <b>90.1</b> |

Table 5.20: Comparison of the semantic segmentation results on the Semantic3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

A detailed comparison of classification performance with other existing methods is presented in Table 5.20. As shown, our TaaNet achieves outstanding performance with an overall accuracy (OA) of 93.7%, surpassing most prior approaches. More impressively, our TaaNet attains the highest mF1 score of 83.9%, outperforming the second-best method by a notable margin of 6.7%, which highlights the enhanced capacity of proposed method to maintain balanced per-class performance even in complex large-scale point clouds. In terms of rare class segmentation, our model demonstrates

particularly strong performance with a remarkably high F1 score of 90.1% for the car category. Taken together, these results further confirm that our TaaNet not only preserves high segmentation accuracy across majority classes, but also substantially improves recognition performance for rare categories, especially when it comes to large-scale imbalanced 3D point clouds.

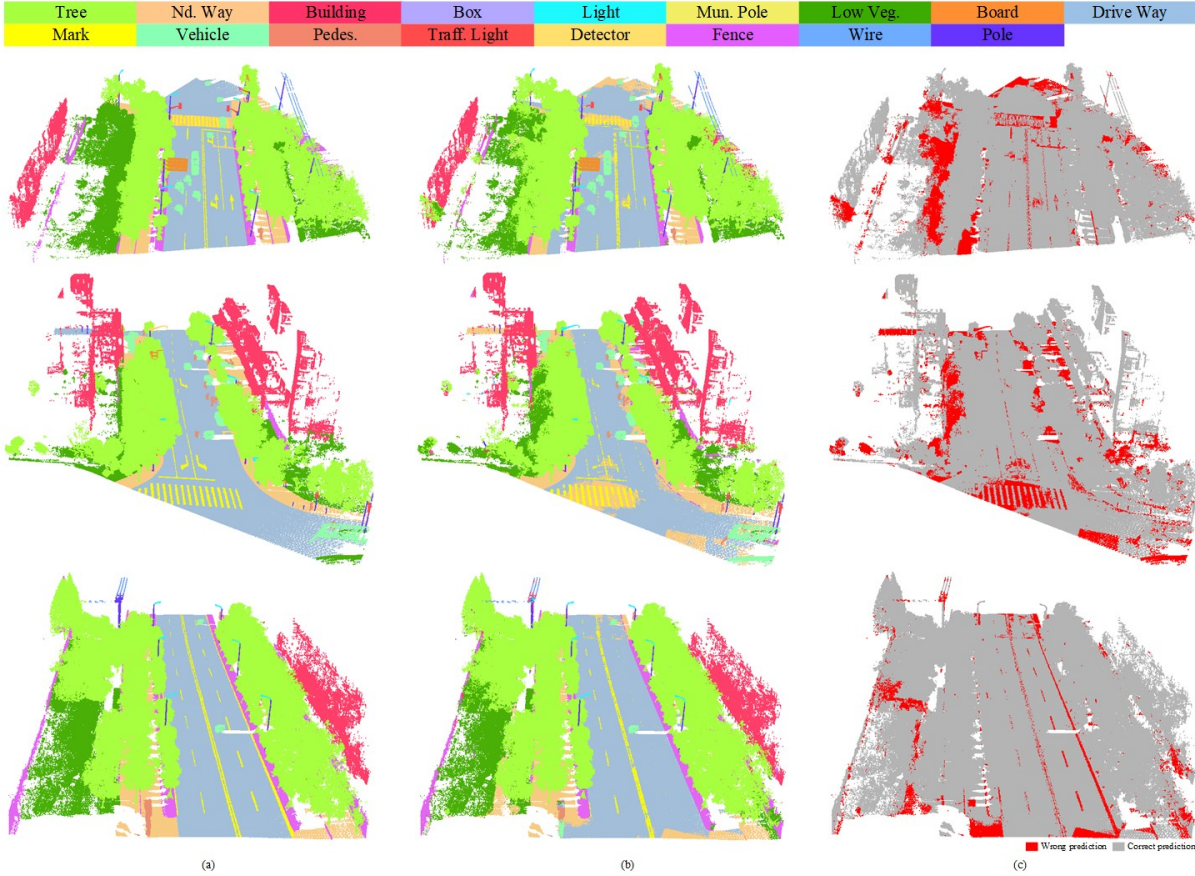


Figure 5.17: Classification results of our TaaNet model on MLS-S in the WHU-Urban3D dataset. Only three selected scenes are shown. Column (a) represents the ground truths. Column (b) presents the classification results using the proposed method. Column (c) presents the corresponding error maps.

## Results on the WHU-Urban3D dataset

In addition to the three datasets discussed above, we further evaluate the proposed TaaNet on the large-scale WHU-Urban3D dataset to assess its scalability and robustness under more complex urban environments. Specifically, we consider both of its subsets: the WHU-Urban3D MLS-S data and the WHU-Urban3D ALS-S data, which correspond to mobile and airborne laser scanning point clouds collected from urban areas in Shanghai, China. The confusion matrices reflecting the classification performance of our method on these two subsets are provided in Table 5.21 and Table 5.23.

Additionally, the qualitative segmentation results and associated error distributions are visualized in Fig. 5.17 and Fig. 5.18. For the WHU-Urban3D MLS-S subset, as illustrated in Fig. 5.17(b), our TaaNet demonstrates a strong capacity to recognize small-scale objects such as lights, boards, and detectors. However, misclassifications are notably observed between the road mark and drive way classes. Such confusion is primarily due to the intrinsic semantic and spatial overlap between these categories, which are difficult to distinguish based solely on geometric descriptors. While intensity information serves as the dominant cue for differentiation in these cases, it often proves inadequate to resolve the ambiguity between these two classes. Furthermore, the overlay of road markings modifies the local surface characteristics in these transitional zones, further contributing to geometric similarity and increased confusion between road mark and non-drive way classes.

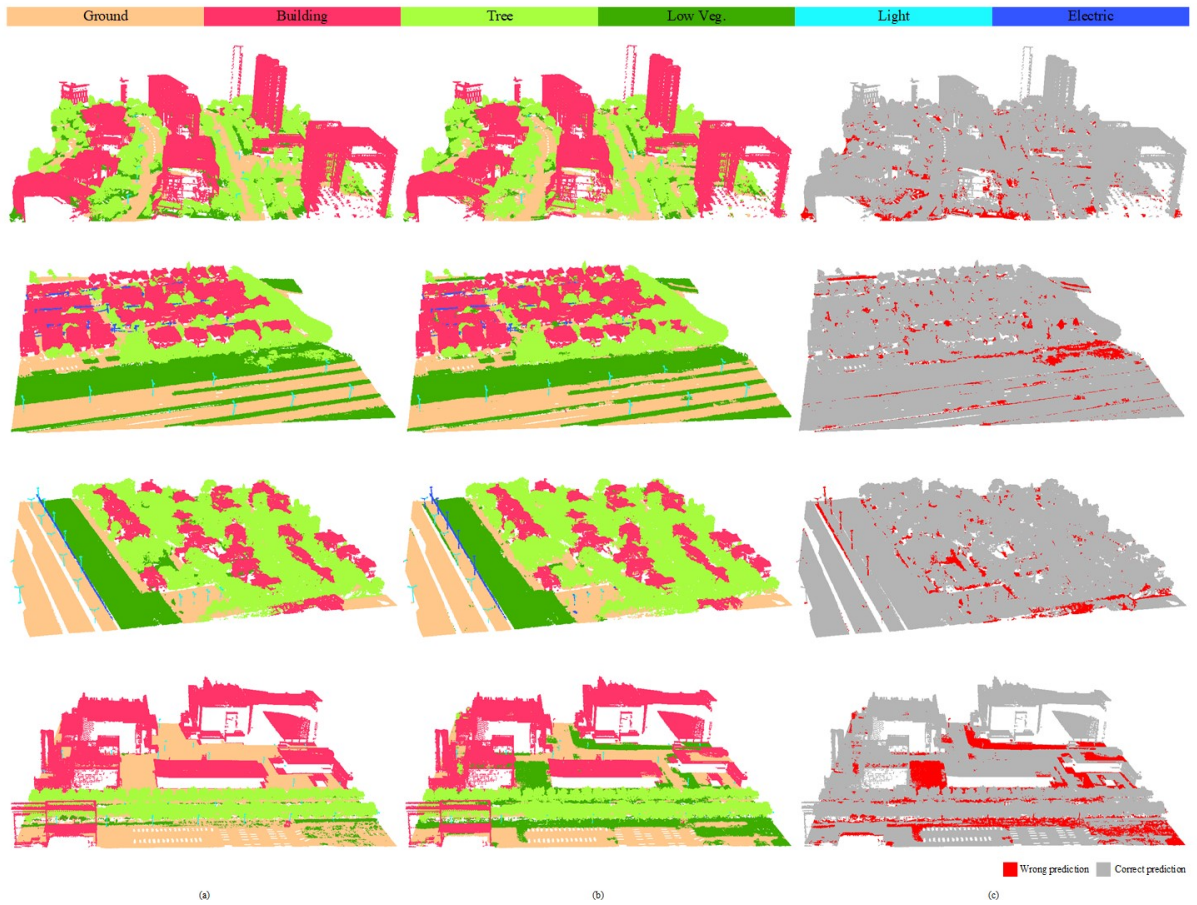


Figure 5.18: Classification results of our TaaNet model on ALS-S in the WHU-Urban3D dataset. Only four selected scenes are shown. Column (a) represents the ground truths. Column (b) presents the classification results using the proposed method. Column (c) presents the corresponding error maps.

| Class        | Tree         | Nd.<br>Way   | Building     | Box          | Light        | Mun.<br>Pole | Low<br>Veg.  | Board        | Drive<br>Way | Mark         | Vehicle      | Pedes.       | Traff.<br>Light | Detector     | Fence        | Wire  | Pole         | Precision |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------------|--------------|--------------|-------|--------------|-----------|
| Tree         | <b>0.927</b> | 0.000        | 0.006        | 0.007        | 0.056        | 0.038        | 0.090        | 0.117        | 0.000        | 0.000        | 0.000        | 0.019        | 0.038           | 0.021        | 0.001        | 0.005 | 0.032        | 0.981     |
| Nd. Way      | 0.001        | <b>0.796</b> | 0.001        | 0.027        | 0.051        | 0.242        | 0.111        | 0.017        | 0.064        | 0.065        | 0.000        | 0.012        | 0.004           | 0.000        | 0.081        | 0.047 | 0.003        | 0.536     |
| Building     | 0.000        | 0.008        | <b>0.974</b> | 0.000        | 0.000        | 0.000        | 0.003        | 0.014        | 0.000        | 0.000        | 0.001        | 0.001        | 0.000           | 0.000        | 0.011        | 0.002 | 0.003        | 0.930     |
| Box          | 0.000        | 0.000        | 0.001        | <b>0.796</b> | 0.000        | 0.008        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.002        | 0.000           | 0.000        | 0.000        | 0.000 | 0.000        | 0.941     |
| Light        | 0.000        | 0.000        | 0.000        | 0.001        | <b>0.735</b> | 0.000        | 0.000        | 0.019        | 0.000        | 0.000        | 0.000        | 0.000        | 0.001           | 0.197        | 0.000        | 0.035 | 0.010        | 0.612     |
| Mun. Pole    | 0.000        | 0.000        | 0.000        | 0.010        | 0.000        | <b>0.090</b> | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000        | 0.000        | 0.000 | 0.001        | 0.349     |
| Low Veg.     | 0.064        | 0.004        | 0.003        | 0.008        | 0.002        | 0.001        | <b>0.766</b> | 0.002        | 0.001        | 0.000        | 0.004        | 0.022        | 0.001           | 0.000        | 0.322        | 0.002 | 0.005        | 0.432     |
| Board        | 0.000        | 0.000        | 0.000        | 0.000        | 0.004        | 0.000        | 0.000        | <b>0.686</b> | 0.000        | 0.000        | 0.000        | 0.000        | 0.002           | 0.066        | 0.000        | 0.000 | 0.010        | 0.909     |
| Drive Way    | 0.000        | 0.114        | 0.001        | 0.022        | 0.025        | 0.292        | 0.019        | 0.004        | <b>0.921</b> | 0.368        | 0.002        | 0.026        | 0.029           | 0.000        | 0.051        | 0.000 | 0.000        | 0.931     |
| Mark         | 0.000        | 0.017        | 0.000        | 0.016        | 0.004        | 0.270        | 0.000        | 0.004        | 0.012        | <b>0.563</b> | 0.000        | 0.005        | 0.001           | 0.000        | 0.001        | 0.000 | 0.000        | 0.735     |
| Vehicle      | 0.000        | 0.000        | 0.005        | 0.028        | 0.000        | 0.005        | 0.001        | 0.000        | 0.001        | 0.000        | <b>0.986</b> | 0.041        | 0.000           | 0.000        | 0.000        | 0.036 | 0.000        | 0.952     |
| Pedes.       | 0.000        | 0.001        | 0.005        | 0.030        | 0.000        | 0.004        | 0.000        | 0.000        | 0.000        | 0.000        | 0.005        | <b>0.836</b> | 0.000           | 0.000        | 0.000        | 0.000 | 0.002        | 0.876     |
| Traff. Light | 0.000        | 0.000        | 0.000        | 0.000        | 0.011        | 0.000        | 0.000        | 0.021        | 0.000        | 0.000        | 0.000        | 0.000        | <b>0.836</b>    | 0.024        | 0.000        | 0.000 | 0.004        | 0.867     |
| Detector     | 0.000        | 0.000        | 0.000        | 0.000        | 0.035        | 0.000        | 0.000        | 0.001        | 0.000        | 0.000        | 0.000        | 0.000        | 0.040           | <b>0.617</b> | 0.000        | 0.000 | 0.002        | 0.901     |
| Fence        | 0.007        | 0.060        | 0.005        | 0.048        | 0.008        | 0.050        | 0.010        | 0.053        | 0.002        | 0.005        | 0.002        | 0.036        | 0.031           | 0.001        | <b>0.531</b> | 0.016 | 0.009        | 0.888     |
| Wire         | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000        | 0.000           | 0.000        | <b>0.837</b> | 0.000 | 0.002        | 0.609     |
| Pole         | 0.000        | 0.000        | 0.000        | 0.009        | 0.009        | 0.001        | 0.000        | 0.063        | 0.000        | 0.000        | 0.000        | 0.001        | 0.018           | 0.073        | 0.000        | 0.021 | <b>0.917</b> | 0.905     |
| Recall       | 0.927        | 0.796        | 0.974        | 0.796        | 0.735        | 0.090        | 0.766        | 0.686        | 0.921        | 0.563        | 0.986        | 0.836        | 0.836           | 0.617        | 0.531        | 0.837 | 0.917        | 0.852     |
| F1           | 0.953        | 0.641        | 0.951        | 0.863        | 0.668        | 0.143        | 0.552        | 0.782        | 0.926        | 0.638        | 0.968        | 0.856        | 0.851           | 0.732        | 0.664        | 0.705 | 0.911        | 0.753     |

Table 5.21: Confusion matrix of our TaaNet on MLS-S in the WHU-Urban3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

| Method        | F1 Score    |             |             |             |             |              |             |             |              |             |             |             |                 |             |             |             |             | OA          |             | rmF1        |
|---------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|-------------|-------------|-----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|               | Tree        | Nd.<br>Way  | Building    | Box         | Light       | Mun.<br>Pole | Low<br>Veg. | Board       | Drive<br>Way | Mark        | Vehicle     | Pedes.      | Traff.<br>Light | Detector    | Fence       | Wire        | Pole        |             |             |             |
| PointNet++    | 93.9        | 63.0        | 80.9        | 0.1         | 52.2        | 0.0          | 38.3        | 1.8         | 93.6         | 70.1        | 86.6        | 41.6        | 22.2            | 0.0         | 61.3        | 0.0         | 46.0        | 83.0        | 44.2        | 5.6         |
| MinkowskiNet  | 95.2        | 53.7        | 78.7        | 52.0        | 48.7        | 9.5          | 42.7        | 38.5        | 92.2         | 66.0        | 90.3        | 75.7        | 58.5            | 41.7        | 66.4        | 33.0        | 75.0        | 83.5        | 59.9        | 40.4        |
| DGCNN         | 94.0        | 63.3        | 92.9        | 29.1        | 60.2        | 0.0          | 42.2        | 30.8        | <b>93.9</b>  | 63.8        | 86.9        | 41.2        | 14.5            | 0.0         | 64.9        | 0.0         | 61.7        | 84.3        | 49.4        | 10.9        |
| PACConv       | 94.9        | 62.3        | 93.6        | 62.2        | 56.5        | 0.0          | 43.5        | 45.4        | 93.7         | <b>70.5</b> | 93.8        | 81.6        | 16.0            | 0.0         | <b>68.1</b> | 0.0         | 85.7        | <b>85.3</b> | 56.9        | 19.6        |
| SPVConv       | <b>95.4</b> | 57.9        | 78.1        | 50.1        | 58.7        | <b>16.4</b>  | 44.8        | 33.1        | 91.9         | 65.2        | 91.3        | 66.2        | 67.4            | 53.9        | 67.8        | 53.5        | 75.9        | 83.8        | 62.8        | 46.9        |
| TaaNet (Ours) | 95.3        | <b>64.1</b> | <b>95.1</b> | <b>86.3</b> | <b>66.8</b> | 14.3         | <b>55.2</b> | <b>78.2</b> | 92.6         | 63.8        | <b>96.8</b> | <b>85.6</b> | <b>85.1</b>     | <b>73.2</b> | 66.4        | <b>70.5</b> | <b>91.1</b> | 85.2        | <b>75.3</b> | <b>64.7</b> |

Table 5.22: Comparison of the semantic segmentation results on MLS-S in the WHU-Urban3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

To further assess the comparative performance, we conduct experiments against other representative baseline methods on the WHU-Urban3D MLS-S subset. The quantitative results are presented in Table 5.22. With an OA of 85.2%, our method demonstrates competitive performance relative to other baseline methods. Although the PAConv slightly surpasses our TaaNet in OA with a score of 85.3%, our method significantly outperforms others in terms of rare class segmentation, as reflected by the rmF1 score. Specifically, the TaaNet attains a rmF1 score of 64.7%, outperforming the second-best method (SPVConv) by a substantial margin of 17.8%. Such a performance gap highlights the effectiveness of our model in handling severe class imbalance, a key limitation of large-scale urban datasets such as the WHU-Urban3D. In particular, the TaaNet achieves strong per-class F1 scores for individual rare classes such as boxes, traffic lights, and detectors, with 86.3%, 85.1%, and 73.2%, respectively. However, a performance limitation of our model is observed for the municipal pole category, where the F1 score drops to 14.3%. This underperformance can be attributed to the extremely sparse point density associated with each individual instance, as well as the lack of distinctive structural patterns in densely cluttered urban scenes.

| Class    | Ground       | Building     | Tree         | Low Veg.     | <u>Light</u> | Electric     | Precision |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|-----------|
| Ground   | <b>0.763</b> | 0.027        | 0.012        | 0.208        | 0.019        | 0.001        | 0.863     |
| Building | 0.065        | <b>0.953</b> | 0.004        | 0.030        | 0.106        | 0.104        | 0.901     |
| Tree     | 0.012        | 0.010        | <b>0.898</b> | 0.049        | 0.192        | 0.028        | 0.943     |
| Low Veg. | 0.160        | 0.010        | 0.086        | <b>0.713</b> | 0.031        | 0.004        | 0.571     |
| Light    | 0.000        | 0.000        | 0.000        | 0.000        | <b>0.578</b> | 0.006        | 0.768     |
| Electric | 0.000        | 0.001        | 0.000        | 0.000        | 0.075        | <b>0.857</b> | 0.938     |
| Recall   | 0.763        | 0.953        | 0.898        | 0.713        | 0.578        | 0.857        | 0.840     |
| F1       | 0.810        | 0.926        | 0.920        | 0.634        | 0.659        | 0.896        | 0.808     |

Table 5.23: Confusion matrix of our TaaNet on ALS-S in the WHU-Urban3D dataset. Precision, recall and F1 score are reported for each class. Note that the column represents the predicted label and the row represents the true label, the rare categories in the dataset are marked in underlined text.

For the WHU-Urban3D ALS-S subset, the classification results are visualized in Fig. 5.18(b), demonstrating that our TaaNet produces accurate predictions for the majority of semantic classes, with smooth and well-defined boundaries between distinct objects. Nonetheless, confusion is frequently observed between ground and low vegetation classes, due to their highly similar surface structures and point distributions. Since 3D coordinate information alone lacks sufficient discriminative power in this context, integrating intensity data could serve as an effective means to provide complementary cues. The quantitative comparisons with baseline methods are listed in Table 5.24. Our method demonstrates superior performance by achieving an mF1 score of 80.8% and an OA of 84.0%, outperforming competing models on the ALS-S subset. Notably, our model shows excep-

tional recognition ability for the light category, obtaining an F1 score of 65.9%, which underscores its capability to capture contextual and geometric patterns associated with rare and spatially sparse objects. However, some lights are misclassified due to occlusion or truncation by nearby structures such as trees or buildings. In particular, when only partial points from the light heads are preserved, they are sometimes incorrectly grouped into the tree class, making it difficult to learn the complete structural characteristics of light objects. Moreover, the similar geometric profiles between light and electric pole classes further contribute to the misclassification between these two categories.

| Method        | F1 Score    |             |             |             |              |             | OA          | mF1         | rmF1        |
|---------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------|
|               | Ground      | Building    | Tree        | Low Veg.    | <u>Light</u> | Electric    |             |             |             |
| PointNet++    | 72.4        | 72.6        | 80.7        | 39.5        | 0.0          | 0.0         | 70.0        | 44.2        | 0.0         |
| MinkowskiNet  | 75.4        | 88.0        | 91.0        | 53.2        | 13.1         | 62.6        | 79.3        | 63.9        | 13.1        |
| DGCNN         | 77.8        | 87.8        | 91.5        | 56.7        | 13.2         | <b>92.1</b> | 81.1        | 69.8        | 13.2        |
| PACConv       | 78.5        | 89.1        | <b>92.5</b> | <b>64.6</b> | 36.0         | 91.4        | 82.7        | 75.3        | 36.0        |
| SPVConv       | 74.4        | 87.6        | 90.7        | 49.8        | 30.9         | 69.2        | 78.4        | 67.1        | 30.9        |
| TaaNet (Ours) | <b>81.0</b> | <b>92.6</b> | 92.0        | 63.4        | <b>65.9</b>  | 89.6        | <b>84.0</b> | <b>80.8</b> | <b>65.9</b> |

Table 5.24: Comparison of the semantic segmentation results on ALS-S in the WHU-Urban3D dataset using our TaaNet and other baseline methods. (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

## Evaluation of generalization ability

Building upon the cross-city semantic annotations provided by the WHU-Urban3D dataset, we conduct a further investigation into the generalization capabilities of our proposed method in comparison with several baseline models. To this end, all models are trained on the WHU-Urban3D MLS-S training subset and evaluated on the WHU-Urban3D MLS-W subset, which represent two distinct urban environments, namely Shanghai and Wuhan, respectively. As the WHU-Urban3D MLS-W subset does not contain any annotated points for the wire category, this class is excluded from the evaluation process. The quantitative results presented in Table 5.25 demonstrate that all methods exhibit a notable decline in performance when generalized to point clouds originating from different urban environments. Such performance degradation can be attributed to inherent variations in urban spatial patterns, including differences in infrastructure design, architectural characteristics, and road layout, which collectively affect the spatial distribution and geometric structure of point clouds. Data-driven models trained on limited spatial distributions often struggle to transfer semantic understanding to unseen urban contexts, due to the absence of diverse training examples that capture such variability. Although the SPVConv achieves superior OA, our

model demonstrates particular strength in the segmentation of rare classes, which are typically more sensitive to cross-domain variations. Therefore, when combined with ensemble learning techniques, our method shows substantial potential to further improve generalization performance on these underrepresented classes.

| Method        | F1 Score    |             |             |             |             |                      |             |             |              |             |             |             |                         |                 |             |      |             | OA          | mF1         | rmF1        |
|---------------|-------------|-------------|-------------|-------------|-------------|----------------------|-------------|-------------|--------------|-------------|-------------|-------------|-------------------------|-----------------|-------------|------|-------------|-------------|-------------|-------------|
|               | Tree        | Nd.<br>Way  | Building    | <u>Box</u>  | Light       | <u>Mun.<br/>Pole</u> | Low<br>Veg. | Board       | Drive<br>Way | Mark        | Vehicle     | Pedes.      | <u>Traff.<br/>Light</u> | <u>Detector</u> | Fence       | Wire | Pole        |             |             |             |
| PointNet++    | 52.4        | 23.7        | 13.5        | 0.0         | 10.5        | 0.0                  | 40.6        | 0.1         | 54.1         | 25.7        | 30.9        | 5.6         | 10.1                    | 0.0             | 19.0        | -    | 26.5        | 41.4        | 19.5        | 2.5         |
| MinkowskiNet  | 51.1        | 26.7        | 12.7        | 8.4         | 16.3        | 7.7                  | 29.8        | 17.4        | 84.0         | 34.7        | 55.7        | 10.8        | 37.4                    | 10.6            | 42.3        | -    | <b>48.4</b> | 56.5        | 30.9        | 16.0        |
| DGCNN         | 54.5        | 22.8        | 17.4        | 0.0         | 21.7        | 0.0                  | 40.5        | 27.9        | 72.3         | 10.4        | 54.4        | 3.5         | 20.2                    | 0.0             | 34.2        | -    | 36.3        | 49.6        | 26.0        | 5.0         |
| PAConv        | 55.4        | 29.2        | 14.4        | 3.2         | 23.4        | 0.0                  | 49.9        | 19.9        | 75.1         | 40.5        | 91.2        | 31.6        | 17.4                    | 0.0             | 52.1        | -    | 37.6        | 55.7        | 33.8        | 5.2         |
| SPVConv       | 53.3        | <b>34.6</b> | 12.1        | 14.3        | 19.5        | 9.1                  | 38.2        | 30.0        | <b>87.0</b>  | 44.2        | 60.8        | 5.5         | 36.3                    | 11.6            | 56.9        | -    | 41.0        | <b>60.3</b> | 34.7        | 17.8        |
| TaaNet (Ours) | <b>56.3</b> | 23.7        | <b>21.6</b> | <b>40.6</b> | <b>26.2</b> | <b>31.4</b>          | <b>57.4</b> | <b>41.5</b> | 46.6         | <b>56.5</b> | <b>93.6</b> | <b>51.6</b> | <b>64.9</b>             | <b>21.7</b>     | <b>70.7</b> | -    | 43.9        | 48.8        | <b>46.8</b> | <b>39.7</b> |

Table 5.25: Generalizability evaluation of our TaaNet and other baseline methods on MLS-W in the WHU-Urban3D dataset (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in per-class F1 score, OA, mF1, and rmF1 are highlighted in bold text.

## Ablation study

To thoroughly validate the effectiveness of the modules proposed in our network architecture, we conduct a series of ablation studies on the Hessigheim 3D dataset in this section. Given that our method is specifically designed to enhance the segmentation of rare categories, we perform targeted experiments focusing on the impact of the target-aware aggregator. Furthermore, recognizing that the weighted loss function introduced in Section 3.4.7 also influences the classification of imbalanced classes, we exclude this commonly adopted strategy in certain ablation settings to independently assess the contribution of each component. The complete experimental results are presented in Table 5.26. Among all model variants, model D, which integrates both the target-aware aggregator and the weighted loss strategy, achieves the best overall performance across all evaluation metrics. Specifically, model D outperforms the baseline model A by margins of 1.4% in OA, 14.0% in mF1, and 56.7% in rmF1, respectively. Notably, even when the weighted loss function is excluded, model B, which solely incorporates the target-aware aggregator, still attains strong segmentation results with an OA of 87.7% and an mF1 of 76.3%. This demonstrates that the target-aware aggregator alone substantially boosts the segmentation of rare categories, as reflected in a 47.4% improvement in rmF1 over the baseline model A. A comparative analysis of models B and C with respect to the baseline model A reveals significant enhancements in mF1, with gains of 11.3% and 11.1%, respectively. These improvements underscore the increased sensitivity of both variants to rare class features. In particular, model B exceeds model A by 47.4% in rmF1, while model C achieves a 42.1% increase. The analysis suggests that while both the target-aware aggregator and the weighted loss strategy enhance rare class segmentation, our aggregator, which is explicitly designed to emphasize features of rare classes, yields a more substantial impact.

| Method | Module       |               | OA          | mF1         | rmF1        |
|--------|--------------|---------------|-------------|-------------|-------------|
|        | Target-aware | Weighted Loss |             |             |             |
|        | Aggregator   | Strategy      |             |             |             |
| A      |              |               | 86.7        | 65.0        | 26.8        |
| B      | ✓            |               | 87.7        | 76.3        | 74.2        |
| C      |              | ✓             | 87.7        | 76.1        | 68.9        |
| D      | ✓            | ✓             | <b>88.2</b> | <b>79.0</b> | <b>83.5</b> |

Table 5.26: Ablation analysis regarding the effect of our target-aware aggregator in the proposed TaaNet on the Hessigheim 3D dataset (values in %). Note that the highest values in OA, mF1, and rmF1 are highlighted in bold text.

| Method        | Params<br>(millions) | Running time<br>(ms/iter) | Memory<br>(GB) |
|---------------|----------------------|---------------------------|----------------|
| PointNet++    | 0.96                 | 498                       | 2.77           |
| MinkowskiNet  | 21.73                | 481                       | 6.56           |
| DGCNN         | 0.98                 | 376                       | 19.59          |
| PAConv        | 11.79                | 441                       | 12.31          |
| SPVConv       | 21.78                | 529                       | 9.06           |
| TaaNet (Ours) | 17.41                | 715                       | 11.07          |

Table 5.27: Comparison of number of parameters, computation time, and GPU memory usage of our TaaNet and other baseline models. It is estimated and averaged on the Hessigheim 3D dataset.

## Complexity and runtime analysis

In order to assess the complexity and training efficiency of our proposed model, we compare the number of parameters, training time, and GPU memory consumption of our TaaNet with those of the aforementioned reference networks, as detailed in Table 5.27. In particular, the running time serves as an indicator of computational efficiency, while GPU memory usage quantifies the hardware resources required for model execution. The number of parameters reflects the overall model complexity, which in turn influences both running time and GPU memory consumption. From the comparison, it is evident that our TaaNet exhibits reduced complexity relative to the MinkowskiNet and the SPVConv, in terms of parameter count. Furthermore, our model demonstrates lower GPU memory usage during training compared to the DGCNN and the PAConv, which indicates improved memory efficiency. Nevertheless, under equivalent implementation settings, the training time of our method is comparatively higher than that of other baseline methods. This additional computational cost primarily stems from the use of attention mechanisms, which require pairwise

correlation computations between elements, rather than relying on parameter-sharing schemes typical of conventional convolutional networks. Despite the increased training time, the enhanced classification performance achieved by our TaaNet suggests that this additional complexity contributes meaningfully to learning capacity. In general, our model maintains a favorable trade-off between the model complexity, computational efficiency and the resulting classification performance.

## Chapter 6

# Conclusion and Outlook

In this chapter, we will present conclusions based on the study conducted in this thesis. The conclusions are organized according to the research questions and specific objectives outlined in Section 1.1 and Section 1.2. Additionally, potential directions for future work are suggested in light of the limitations of the proposed methods.

### 6.1 Conclusion

In this thesis, we focus on the development of an attention-based segmentation framework for point clouds, aiming to achieve accurate semantic prediction in large-scale urban scenes. The motivation behind it lies in the growing need for effective and automated semantic segmentation for point clouds in remote sensing applications, particularly in the context of complex urban environments. Thus, we address several under-explored urban challenges, including class diversity, fine-grained boundary ambiguity, and class imbalance. The main contributions of this thesis are summarized as follows:

- (1) To extract robust and discriminative features for semantic segmentation in complex urban scenes, we design a novel attention-based framework that focuses on key regions in large-scale point clouds while suppressing noise and outliers. The local aggregation module, built upon self-attention operators, redundant positional encoding, and multi-head structure, provides a solid foundation for effectively capturing local features in highly intricate areas. Meanwhile, the hierarchical network architecture enables progressive feature aggregation across different spatial scales, establishing global relationships between point-wise features. Furthermore, skip connections in the hierarchical design facilitate the feature fusion across layers, which preserves

critical semantic information from early stages. The synergy of these components ensures high-accuracy semantic segmentation even in large-scale urban point clouds characterized by varying point densities, significant noise and outliers.

- (2) In addressing the challenge of fine-grained boundary structures in urban point clouds, we introduce a local context embedding method that integrates original surface features into the attention-based network architecture. Multi-scale surface descriptors are adopted as extended variants to enrich the backbone network with comprehensive geometric context. Moreover, the loosely coupled design of surface feature extractor offers a flexible solution for fine-grained segmentation tasks. This integration enables the attention network to more effectively capture subtle local geometric variations, which is particularly beneficial for distinguishing fine-grained boundaries with limited inter-class distances in complex urban scenes.
- (3) To address the under-representation of rare classes caused by inherent data imbalance in urban scenes, we propose a contextual joint augmentation strategy to enhance the representation of rare categories in point cloud segmentation tasks. This plug-and-play method integrates two complementary augmentation branches at the scene and class levels, diversifying the spatial distribution of rare classes and their surrounding contexts. Additionally, a dynamic joint augmentation, coupled with a context-guided selection module, promotes interaction between rare targets and surrounding features while maintaining semantic consistency. By jointly improving the diversity of both rare class representations and their contextual cues, the proposed augmentation approach significantly improves segmentation performance for rare classes.
- (4) Apart from the augmentation strategy, we further investigate solutions that leverage the unique structural characteristics of rare classes. A target-aware attention network is introduced to capture the spatial discrepancies between individual instances of rare categories and optimize their feature representations in the network. An actively supervised framework, namely the target-aware aggregator, extracts structural features of rare targets and embeds their corresponding contextual constraints into point-wise attention, which encapsulates the distinctive topological cues of targets at the instance level. As the core in the aggregator, the target focus aggregation module incorporates the advantages of both local focus attention and instance-level features, enabling adaptive modulation of attention weights for rare categories. This allows our method to achieve significant improvements in segmenting both majority and minority classes in complex urban point clouds. This enables the proposed method to achieve excellent performance in segmenting both majority and rare classes in complex urban point clouds, with particularly notable improvements for rare classes.

## 6.2 Outlook

Building upon the research outcome of this thesis, we discuss the limitations of the proposed methods in practical applications and suggest potential directions for future research.

**Optimization of network architecture** While our attention-based backbone demonstrates strong performance on several benchmark datasets, point-based methods remain computationally inefficient during both training and inference and are sensitive to sub-region boundary divisions. Currently, several deep learning approaches are working on clusters of points instead of directly on discrete points, which significantly reduce the input size and improve computational efficiency while preserving natural class boundaries. Future research could focus on the development of a hybrid network architecture that integrates both point-wise and cluster-wise representations, aiming to maintain segmentation accuracy while allowing flexible trade-offs between computational cost and runtime efficiency according to specific application requirements. In addition, the proposed target-aware aggregator still suffers from uncertain errors of self-supervised signals, which are largely influenced by the spatial distribution of rare objects. Meanwhile, the simplified description of target contours makes it only applicable to approximate cuboid instances. In future work, we will enhance the target-aware aggregator by exploring more robust and fine-grained representations of target topologies, enabling better adaptation to diverse target shapes and distributions.

**Enhancement of augmentation strategy** Although our contextual joint augmentation strategy build semantic consistency between rare categories and their surrounding sub-scenes to a certain extent, the complexity of urban environments still limits the generation of high-quality augmented data. Recent advances in generative models, such as diffusion models and generative adversarial networks (GANs), have shown promising results in image and 3D object synthesis. Building on this, future research can explore two complementary directions: generateting semantically reasonable rare objects in complex geospatial contexts, and developing point cloud completion methods for rare classes to mitigate issues caused by occlusion. Such enhancements to augmentation strategies are expected to improve the segmentation performance of rare classes and promote the generalization ability of our framework.

**Fusion of multi-modality data** Our current research focuses merely on LiDAR point clouds, but multi-modality data fusion is expected to enhance the perception of models by leveraging the complementary characteristics of different sensor modalities. Although a small number of benchmark datasets provide point-wise texture characteristics registered with RGB images, the advancement of multi-sensor technologies is rapidly increasing the demand for 3D segmentation algorithms using multi-modality data. Thus, the specialized alignment and fusion of various data modalities

to provide a comprehensive understanding of urban scenes remains a key issue worthy of further investigation. Furthermore, vision language models (VLMs) have demonstrated remarkable capabilities in the field of computer vision. Aligning the rich semantic signals from pre-trained language models with 3D point cloud processing represents a promising and exciting future direction.

# Bibliography

- Aksoy, E. E., Baci, S., and Cavdar, S. (2020). Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *2020 IEEE intelligent vehicles symposium (IV)*, pages 926–932. IEEE.
- Alnaggar, Y. A., Afifi, M., Amer, K., and ElHelw, M. (2021). Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1800–1809.
- Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., and Marlet, R. (2023). Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5240–5250.
- Angelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., and Ng, A. (2005). Discriminative learning of markov random fields for segmentation of 3d scan data. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 169–176. IEEE.
- Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122.
- Bassier, M. and Vergauwen, M. (2020). Unsupervised reconstruction of building information modeling wall objects from point cloud data. *Automation in construction*, 120:103338.
- Bayrak, O. C., Ma, Z., Farella, E. M., Remondino, F., and Uzar, M. (2024). Estate: a large dataset of under-represented urban objects for 3d point cloud classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:25–32.
- Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307.
- Belton, D. and Lichti, D. D. (2006). Classification and segmentation of terrestrial laser scanner point clouds using local variance information. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 36(5):44–49.
- Besl, P. J. and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Transactions on pattern analysis and machine intelligence*, 10(2):167–192.
- Blomley, R., Weinmann, M., Leitloff, J., and Jutzi, B. (2014). Shape distribution features for point cloud analysis—a geometric histogram approach on multiple scales. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:9–16.
- Boulch, A. (2020). Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*,

- 88:24–34.
- Boulch, A., Guerry, J., Le Saux, B., and Audebert, N. (2018). Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 71:189–198.
- Brodu, N. and Lague, D. (2012). 3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS journal of photogrammetry and remote sensing*, 68:121–134.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.
- Chehata, N., Guo, L., and Mallet, C. (2009). Airborne lidar feature selection for urban classification using random forests. In *Laserscanning*.
- Chen, S., Liu, B., Feng, C., Vallespi-Gonzalez, C., and Wellington, C. (2021). 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine*, 38(1):68–86.
- Chen, S., Yang, B., Xia, Y., Cheng, M., Shen, S., and Wang, C. (2024). Bridging lidar gaps: a multi-lidars domain adaptation dataset for 3d semantic segmentation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 650–658.
- Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299.
- Cheng, B., Schwing, A., and Kirillov, A. (2021a). Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875.
- Cheng, R., Razani, R., Taghavi, E., Li, E., and Liu, B. (2021b). 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12547–12556.
- Choy, C., Gwak, J., and Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084.
- Cordonnier, J.-B., Loukas, A., and Jaggi, M. (2020). On the relationship between self-attention and convolutional layers. In *Eighth International Conference on Learning Representations-ICLR 2020*.
- Cui, Y., Chen, R., Chu, W., Chen, L., Tian, D., Li, Y., and Cao, D. (2021). Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739.
- Demantké, J., Mallet, C., David, N., and Vallet, B. (2011). Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W12:97–102.
- Deschaud, J.-E. and Goulette, F. (2010). A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In *3DPVT*.
- Dong, Z., Yang, B., Hu, P., and Scherer, S. (2018). An efficient global energy optimization approach for robust 3d plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*,

- 137:112–133.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Duan, Z., Shao, J., Zhang, M., Zhang, J., and Zhai, Z. (2024). A small-object-detection algorithm based on lidar point-cloud clustering for autonomous vehicles. *Sensors*, 24(16):5423.
- Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658.
- Engel, N., Belagiannis, V., and Dietmayer, K. (2021). Point transformer. *IEEE access*, 9:134826–134840.
- Engelmann, F., Kontogianni, T., Hermans, A., and Leibe, B. (2017). Exploring spatial context for 3d semantic segmentation of point clouds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 716–724.
- Engelmann, F., Kontogianni, T., and Leibe, B. (2020). Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9463–9469. IEEE.
- Engelmann, F., Kontogianni, T., Schult, J., and Leibe, B. (2018). Know what your neighbors do: 3d semantic segmentation of point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96(34), pages 226–231.
- Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., and Wang, F.-Y. (2021). Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14504–14513.
- Fang, J., Zuo, X., Zhou, D., Jin, S., Wang, S., and Zhang, L. (2021). Lidar-aug: A general rendering-based augmentation framework for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4710–4720.
- Feng, C., Taguchi, Y., and Kamat, V. R. (2014). Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225. IEEE.
- Feng, Y., Zhang, Z., Zhao, X., Ji, R., and Gao, Y. (2018). Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 264–272.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.
- Golovinskiy, A. and Funkhouser, T. (2009). Min-cut based segmentation of point clouds. In *2009 IEEE 12th*

- International Conference on Computer Vision Workshops, ICCV Workshops*, pages 39–46. IEEE.
- Graham, B. (2015). Sparse 3d convolutional neural networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 150.1–150.9. BMVA Press.
- Graham, B., Engelcke, M., and Van Der Maaten, L. (2018). 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232.
- Grandio, J., Riveiro, B., Soilán, M., and Arias, P. (2022). Point cloud semantic segmentation of complex railway environments using deep learning. *Automation in Construction*, 141:104425.
- Griffiths, D. and Boehm, J. (2019). Weighted point cloud augmentation for neural network training data class-imbalance. *arXiv preprint arXiv:1904.04094*.
- Gumhold, S., Wang, X., MacLeod, R. S., et al. (2001). Feature extraction from point clouds. In *IMR*, pages 293–305.
- Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., and Hu, S.-M. (2021). Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199.
- Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364.
- Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., and Pollefeys, M. (2017). Semantic3d.net: A new large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1:91–98.
- Hackel, T., Wegner, J. D., and Schindler, K. (2016). Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3:177–184.
- Han, X., Liu, C., Zhou, Y., Tan, K., Dong, Z., and Yang, B. (2024). Whu-urban3d: An urban scene lidar point cloud dataset for semantic instance segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 209:500–513.
- Han, X.-F., Jin, Y.-F., Cheng, H.-X., and Xiao, G.-Q. (2022). Dual transformer for point cloud analysis. *IEEE Transactions on Multimedia*, 25:5638–5648.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., and Markham, A. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117.
- Huang, R., Hong, D., Xu, Y., Yao, W., and Stilla, U. (2019). Multi-scale local context embedding for lidar point cloud classification. *IEEE Geoscience and Remote Sensing Letters*, 17(4):721–725.
- Huang, R., Xu, Y., Hong, D., Yao, W., Ghamisi, P., and Stilla, U. (2020). Deep point embedding for urban classification using als point clouds: A new perspective from local to global. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:62–81.
- Huang, R., Xu, Y., and Stilla, U. (2021). Granet: Global relation-aware attentional network for semantic segmentation of als point clouds. *ISPRS Journal of photogrammetry and remote sensing*, 177:1–20.

- Huang, Y., Zheng, W., Zhang, Y., Zhou, J., and Lu, J. (2023). Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9223–9232.
- Hui, L., Yuan, J., Cheng, M., Xie, J., Zhang, X., and Yang, J. (2021). Superpoint network for point cloud oversegmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5510–5519.
- Jiang, L., Zhao, H., Liu, S., Shen, X., Fu, C.-W., and Jia, J. (2019). Hierarchical point-edge interaction network for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10433–10441.
- Jiang, M., Wu, Y., Zhao, T., Zhao, Z., and Lu, C. (2018). Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*.
- Jung, H., Choi, H.-S., and Kang, M. (2021). Boundary enhancement semantic segmentation for building extraction from remote sensed image. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12.
- Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41.
- Kim, H., Sohn, G., et al. (2010). 3d classification of power-line scene from airborne laser scanning data using random forests. *Int. Arch. Photogramm. Remote Sens*, 38(1999):126–32.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026.
- Klokov, R. and Lempitsky, V. (2017). Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE international conference on computer vision*, pages 863–872.
- Kong, L., Liu, Y., Chen, R., Ma, Y., Zhu, X., Li, Y., Hou, Y., Qiao, Y., and Liu, Z. (2023). Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 228–240.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., and Ledoux, H. (2021). The hessigheim 3d (h3d) benchmark on semantic segmentation of high-resolution 3d point clouds and textured meshes from uav lidar and multi-view-stereo. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 1:100001.
- Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., and Jia, J. (2022). Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8500–8509.
- Lalonde, J.-F., Unnikrishnan, R., Vandapel, N., and Hebert, M. (2005). Scale selection for classification of point-sampled 3d surfaces. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 285–292. IEEE.
- Landrieu, L. and Boussaha, M. (2019). Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

- pages 7440–7449.
- Landrieu, L. and Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567.
- Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., and Felsberg, M. (2017). Deep projective 3d semantic segmentation. In *Computer Analysis of Images and Patterns: 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*, pages 95–107. Springer.
- Lee, I. and Schenk, T. (2002). Perceptual organization of 3d surface points. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):193–198.
- Lei, H., Akhtar, N., and Mian, A. (2019). Octree guided cnn with spherical kernels for 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9631–9640.
- Lei, H., Akhtar, N., and Mian, A. (2021). Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3664–3680.
- Li, H., Xiong, P., An, J., and Wang, L. (2018a). Pyramid attention network for semantic segmentation. In *29th British Machine Vision Conference, BMVC 2018*.
- Li, J., Chen, B. M., and Lee, G. H. (2018b). So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406.
- Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. (2018c). Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31.
- Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M. A., Cao, D., and Li, J. (2020). Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432.
- Liang, Z., Li, Z., Xu, S., Tan, M., and Jia, K. (2021). Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2783–2792.
- Liang, Z., Yang, M., Deng, L., Wang, C., and Wang, B. (2019). Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8152–8158. IEEE.
- Lin, D., Dong, Z., Zhang, X., and Maas, H.-G. (2019). Unsupervised window extraction from photogrammetric point clouds with thermal attributes. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:45–51.
- Lin, H.-I. and Nguyen, M. C. (2020). Boosting minority class prediction on imbalanced point cloud data. *Applied Sciences*, 10(3):973.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Lin, Z.-H., Huang, S.-Y., and Wang, Y.-C. F. (2020). Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1800–1809.
- Liu, Y., Chen, R., Li, X., Kong, L., Yang, Y., Xia, Z., Bai, Y., Zhu, X., Ma, Y., Li, Y., et al. (2023). Uniseg:

- A unified multi-modal lidar segmentation network and the openpcseg codebase. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21662–21673.
- Liu, Y., Fan, B., Xiang, S., and Pan, C. (2019a). Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8895–8904.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.
- Liu, Z., Tang, H., Lin, Y., and Han, S. (2019b). Point-voxel cnn for efficient 3d deep learning. *Advances in neural information processing systems*, 32.
- Ma, X., Qin, C., You, H., Ran, H., and Fu, Y. (2022). Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.
- Ma, Y., Guo, Y., Liu, H., Lei, Y., and Wen, G. (2020). Global context reasoning for semantic segmentation of 3d point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2931–2940.
- Maas, H.-G. and Vosselman, G. (1999). Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):153–163.
- Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE.
- Milioto, A., Vizzo, I., Behley, J., and Stachniss, C. (2019). Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 4213–4220. IEEE.
- Min, C., Xiao, L., Zhao, D., Nie, Y., and Dai, B. (2023). Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. *IEEE Transactions on Intelligent Vehicles*, 9(7):5150–5162.
- Morsdorf, F., Meier, E., Kötz, B., Itten, K. I., Dobbertin, M., and Allgöwer, B. (2004). Lidar-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote sensing of environment*, 92(3):353–362.
- Nagy, B. and Benedek, C. (2018). Real-time point cloud alignment for vehicle localization in a high resolution 3d map. In *Proceedings of the european conference on computer vision (ECCV) workshops*, pages 0–0.
- Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F., and Yang, M.-H. (2021). Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308.
- Nguyen, A. and Le, B. (2013). 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE.
- Niemeyer, J., Rottensteiner, F., and Soergel, U. (2012). Conditional random fields for lidar point cloud classification in complex urban areas. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 1:263–268.

- Niemeyer, J., Rottensteiner, F., and Soergel, U. (2014). Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 87:152–165.
- Pan, Y., Gao, B., Mei, J., Geng, S., Li, C., and Zhao, H. (2020). Semanticpos: A point cloud dataset with large quantity of dynamic instances. In *2020 IEEE intelligent vehicles symposium (IV)*, pages 687–693. IEEE.
- Park, C., Jeong, Y., Cho, M., and Park, J. (2022). Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16949–16958.
- Pauly, M., Keiser, R., and Gross, M. (2003). Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, volume 22(3), pages 281–289. Wiley Online Library.
- Pu, S. and Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):575–584.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.
- Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.
- Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., and Ghanem, B. (2022). Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204.
- Qiu, B., Zhou, Y., Dai, L., Wang, B., Li, J., Dong, Z., Wen, C., Ma, Z., and Yang, B. (2024). Whu-railway3d: A diverse dataset and benchmark for railway point cloud semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*.
- Rabbani, T. and Van Den Heuvel, F. (2005). Efficient hough transform for automatic detection of cylinders in point clouds. *Isprs Wg Iii/3, Iii/4*, 3:60–65.
- Rabbani, T., Van Den Heuvel, F., and Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *International archives of photogrammetry, remote sensing and spatial information sciences*, 36(5):248–253.
- Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. (2019). Stand-alone self-attention in vision models. *Advances in neural information processing systems*, 32.
- Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. (2020). Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186.
- Ren, P. and Xia, Q. (2023). Classification method for imbalanced lidar point cloud based on stack autoencoder. *Electronic Research Archive*, 31(6).
- Riegler, G., Osman Ulusoy, A., and Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586.
- Robert, D., Raguet, H., and Landrieu, L. (2023). Efficient 3d semantic segmentation with superpoint transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17195–17204.

- Robert, D., Raguét, H., and Landrieu, L. (2024). Scalable 3d panoptic segmentation as superpoint graph clustering. In *2024 International Conference on 3D Vision (3DV)*, pages 179–189. IEEE.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer.
- Roynard, X., Deschaud, J.-E., and Goulette, F. (2018). Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557.
- Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE.
- Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941.
- Saltori, C., Galasso, F., Fiameni, G., Sebe, N., Poiesi, F., and Ricci, E. (2023). Compositional semantic mix for domain adaptation in point cloud segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14234–14247.
- Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., and Sattarvand, J. (2024). A comprehensive overview of deep learning techniques for 3d point cloud classification and semantic segmentation. *Machine Vision and Applications*, 35(4):67.
- Schmohl, S. and Sörgel, U. (2019). Submanifold sparse convolutional networks for semantic segmentation of large-scale als point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:77–84.
- Schnabel, R., Wahl, R., and Klein, R. (2007a). Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26(2), pages 214–226. Wiley Online Library.
- Schnabel, R., Wahl, R., and Klein, R. (2007b). Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26(2), pages 214–226. Wiley Online Library.
- Secord, J. and Zakhor, A. (2007). Tree detection in urban regions using aerial lidar and image data. *IEEE Geoscience and Remote Sensing Letters*, 4(2):196–200.
- Shapovalov, R., Velizhev, A., and Barinova, O. (2010). Non-associative markov networks for 3d point cloud classification. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*, pages 103–108.
- Shi, S., Wang, X., and Li, H. (2019). Pointtrcnn: 3d object proposal generation and detection from point cloud. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779.
- Shi, W. and Rajkumar, R. (2020). Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719.
- Simonovsky, M. and Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702.

- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Soilán, M., Truong-Hong, L., Riveiro, B., and Laefer, D. (2018). Automatic extraction of road features in urban environments using dense als data. *International journal of applied earth observation and geoinformation*, 64:226–236.
- Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., and Li, J. (2020). 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 workshops*, pages 202–203.
- Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., and Han, S. (2020). Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer.
- Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data. In *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, volume 36, pages 407–412.
- Tatarchenko, M., Park, J., Koltun, V., and Zhou, Q.-Y. (2018). Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3887–3896.
- Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. (2019). Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420.
- Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part III 11*, pages 356–369. Springer.
- Tóvári, D. and Pfeifer, N. (2005). Segmentation based robust interpolation—a new approach to laser data filtering. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(3/19):79–84.
- Varney, N., Asari, V. K., and Graehling, Q. (2020). 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*, pages 186–187.
- Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
- Vo, A.-V., Truong-Hong, L., Laefer, D. F., and Bertolotto, M. (2015). Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:88–100.
- Vosselman, G., Gorte, B. G., Sithole, G., and Rabbani, T. (2004). Recognising structure in laser scanner point clouds. *International archives of photogrammetry, remote sensing and spatial information sciences*, 46(8):33–38.

- Wang, J. and Shan, J. (2009). Segmentation of lidar point clouds for building extraction. In *American Society for Photogramm. Remote Sens. Annual Conference, Baltimore, MD*, pages 9–13.
- Wang, L., Huang, Y., Hou, Y., Zhang, S., and Shan, J. (2019a). Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10296–10305.
- Wang, P. and Yao, W. (2022). A new weakly supervised approach for als point cloud semantic segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 188:237–254.
- Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578.
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019b). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12.
- Weinmann, M., Jutzi, B., Hinz, S., and Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304.
- Weinmann, M., Jutzi, B., and Mallet, C. (2013). Feature relevance assessment for the semantic interpretation of 3d point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-5/W2:313–318.
- Wu, B., Wan, A., Yue, X., and Keutzer, K. (2018). Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE.
- Wu, W., Qi, Z., and Fuxin, L. (2019). Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630.
- Xiao, A., Guan, D., Zhang, X., and Lu, S. (2023). Domain adaptive lidar point cloud segmentation with 3d spatial consistency. *IEEE Transactions on Multimedia*, 26:5536–5547.
- Xiao, A., Huang, J., Guan, D., Cui, K., Lu, S., and Shao, L. (2022). Polarmix: A general data augmentation technique for lidar point clouds. *Advances in Neural Information Processing Systems*, 35:11035–11048.
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090.
- Xie, S., Gu, J., Guo, D., Qi, C. R., Guibas, L., and Litany, O. (2020a). Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *16th European Conference on Computer Vision, ECCV 2020*, pages 574–591. Springer Science and Business Media Deutschland GmbH.
- Xie, Y., Tian, J., and Zhu, X. X. (2020b). Linking points with labels in 3d: A review of point cloud semantic segmentation. *IEEE Geoscience and remote sensing magazine*, 8(4):38–59.
- Xie, Z., Chen, J., and Peng, B. (2020c). Point clouds learning with attention-based graph convolution networks. *Neurocomputing*, 402:245–255.

- Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., and Pu, S. (2021a). Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16024–16033.
- Xu, M., Ding, R., Zhao, H., and Qi, X. (2021b). Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182.
- Xu, Q., Sun, X., Wu, C.-Y., Wang, P., and Neumann, U. (2020). Grid-gcn for fast and scalable point cloud learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5661–5670.
- Xu, Y., Fan, T., Xu, M., Zeng, L., and Qiao, Y. (2018). Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European conference on computer vision (ECCV)*, pages 87–102.
- Xu, Y., Hoegner, L., Tuttas, S., and Stilla, U. (2017). Voxel-and graph-based point cloud segmentation of 3d scenes using perceptual grouping laws. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:43–50.
- Xue, R., Zhang, X., and Soergel, U. (2022). Urban classification based on top-view point cloud and sar image fusion with swin transformer. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:559–564.
- Yan, X., Zheng, C., Li, Z., Wang, S., and Cui, S. (2020). 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*, pages 5589–5598.
- Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., and Gao, J. (2021). Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34:30008–30022.
- Yi, L., Gong, B., and Funkhouser, T. (2021). Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15363–15373.
- You, H., Feng, Y., Zhao, X., Zou, C., Ji, R., and Gao, Y. (2019). Pvrnet: Point-view relation neural network for 3d shape recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33(01), pages 9119–9126.
- Yousefhussien, M., Kelbe, D. J., Ientilucci, E. J., and Salvaggio, C. (2018). A multi-scale fully convolutional network for semantic labeling of 3d point clouds. *ISPRS journal of photogrammetry and remote sensing*, 143:191–204.
- Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., and Lu, J. (2022). 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*, pages 19313–19322.
- Yue, C., Wang, Y., Tang, X., and Chen, Q. (2022). Drgcnn: Dynamic region graph convolutional neural network for point clouds. *Expert Systems with Applications*, 205:117663.
- Zhang, F., Fang, J., Wah, B., and Torr, P. (2020). Deep fusionnet for point cloud semantic segmentation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020*,

- Proceedings, Part XXIV 16*, pages 644–663. Springer.
- Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., and Bai, X. (2023a). Deep learning-based 3d point cloud classification: A systematic survey and outlook. *Displays*, 79:102456.
- Zhang, J., Zhao, X., Chen, Z., and Lu, Z. (2019). A review of deep learning-based semantic segmentation for point cloud. *IEEE access*, 7:179118–179133.
- Zhang, N., Pan, Z., Li, T. H., Gao, W., and Li, G. (2023b). Improving graph representation for point cloud segmentation via attentive filtering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1244–1254.
- Zhang, R., Li, G., Wunderlich, T., and Wang, L. (2021). A survey on deep learning-based precise boundary recovery of semantic segmentation for images and point clouds. *International journal of applied earth observation and geoinformation*, 102:102411.
- Zhang, X., Lin, D., and Soergel, U. (2025). Target-aware attentional network for rare class segmentation in large-scale lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 220:32–50.
- Zhang, X., Lin, D., Xue, R., and Soergel, U. (2023c). Target-guided learning for rare class segmentation in large-scale urban point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023:1693–1698.
- Zhang, X., Xue, R., Kölle, M., and Soergel, U. (2022a). Point surfel transformer network for semantic segmentation of large-scale als point clouds. In *42. Wissenschaftlich-Technische Jahrestagung der DGPF, 5.-6. Oktober 2022 in Dresden*, pages 290–296. Geschäftsstelle der DGPF.
- Zhang, X., Xue, R., and Soergel, U. (2022b). A two-stage approach for rare class segmentation in large-scale urban point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:329–334.
- Zhao, H., Jiang, L., Fu, C.-W., and Jia, J. (2019). Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5565–5573.
- Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268.
- Zhao, R., Pang, M., and Wang, J. (2018). Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network. *International journal of geographical information science*, 32(5):960–979.
- Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890.
- Zhou, J., Xiong, Y., Chiu, C., Liu, F., and Gong, X. (2023a). Fat: Field-aware transformer for 3d point cloud semantic segmentation. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 660–664. IEEE.
- Zhou, Y., Ji, A., Zhang, L., and Xue, X. (2023b). Sampling-attention deep learning network with transfer learning for large-scale urban point cloud semantic segmentation. *Engineering Applications of Artificial Intelligence*, 117:105554.

- Zhou, Y., Wang, D., Xie, X., Ren, Y., Li, G., Deng, Y., and Wang, Z. (2014). A fast and accurate segmentation method for ordered lidar point cloud of large-scale scenes. *IEEE Geoscience and Remote Sensing Letters*, 11(11):1981–1985.
- Zhou, Z., Zhang, Y., and Foroosh, H. (2021). Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13194–13203.
- Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable detr: Deformable transformers for end-to-end object detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

# Appendix

## A Supplementary Materials from the Attention Network

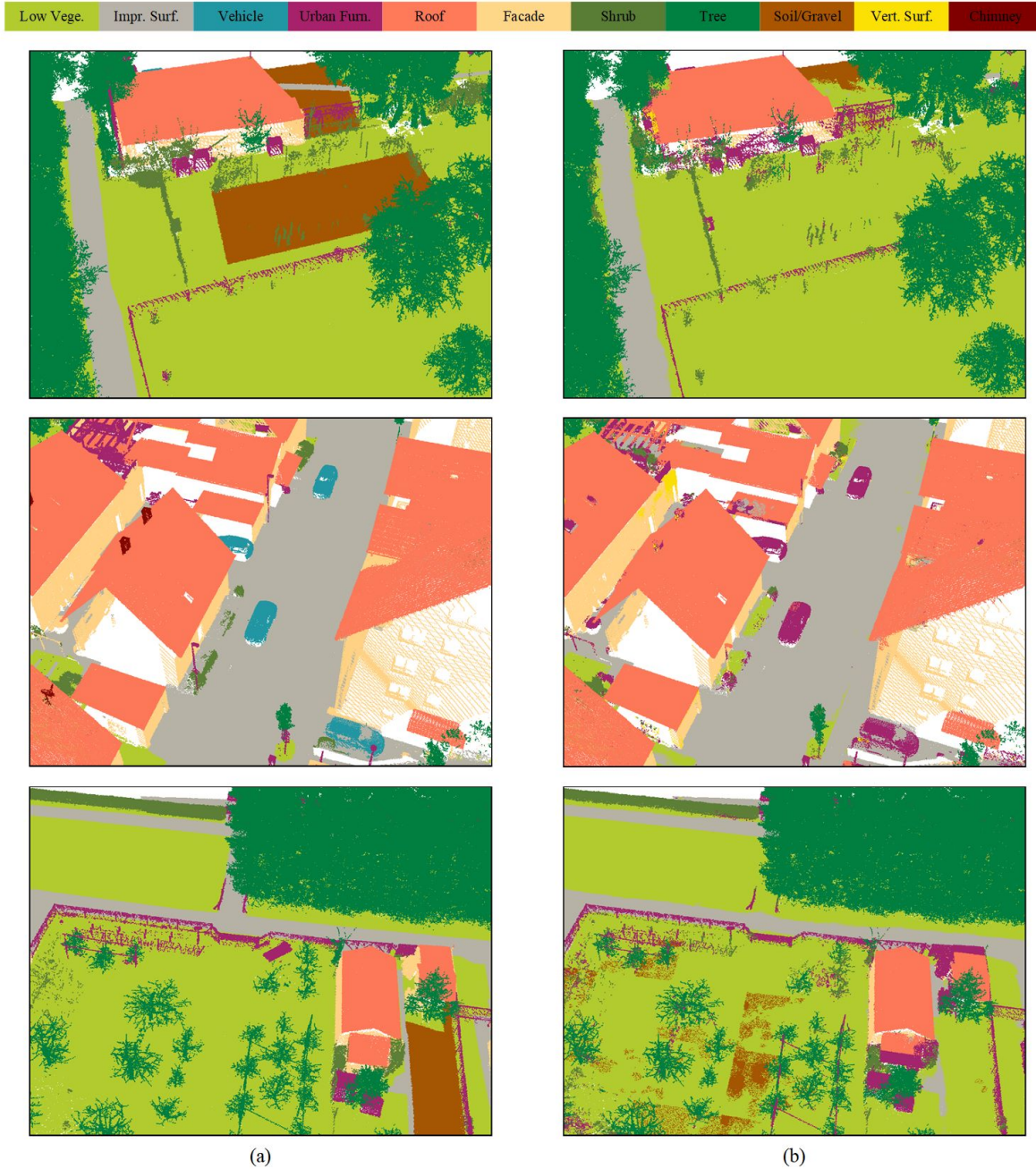


Figure A1: Detailed classification results of the attention network on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

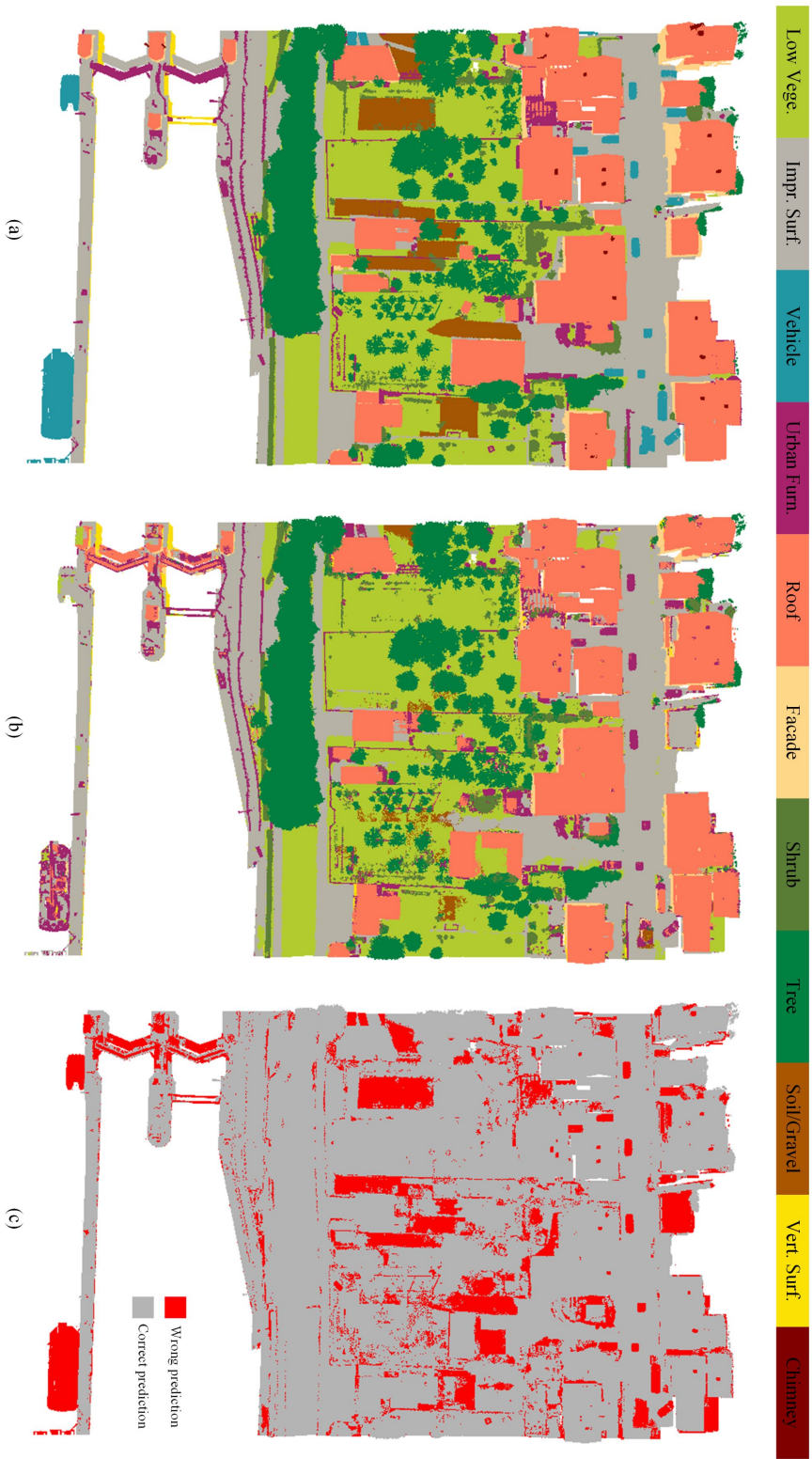


Figure A2: Classification results of the attention network on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

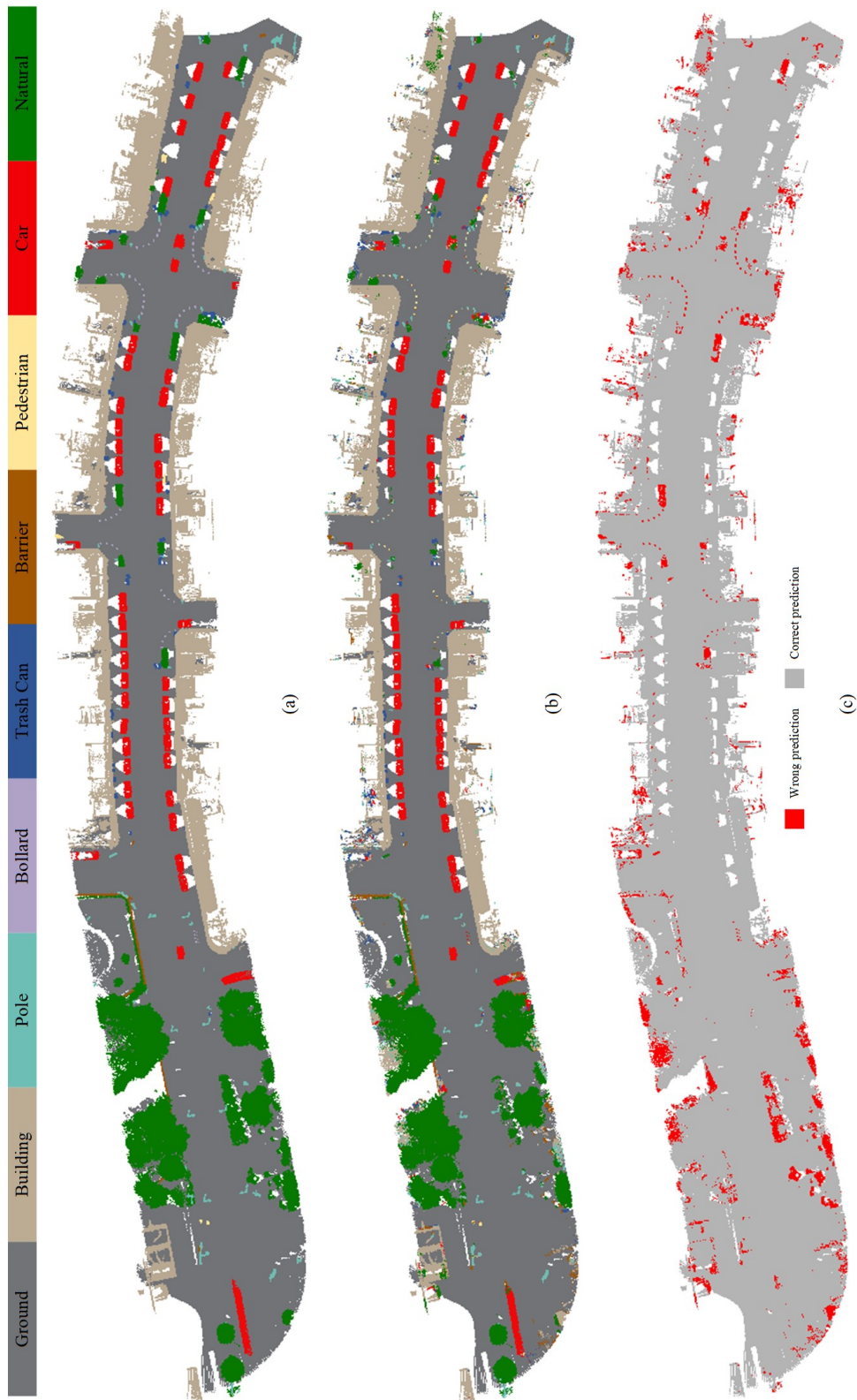


Figure A3: Classification results of the attention network on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## B Supplementary Materials from the Local Context Embedding

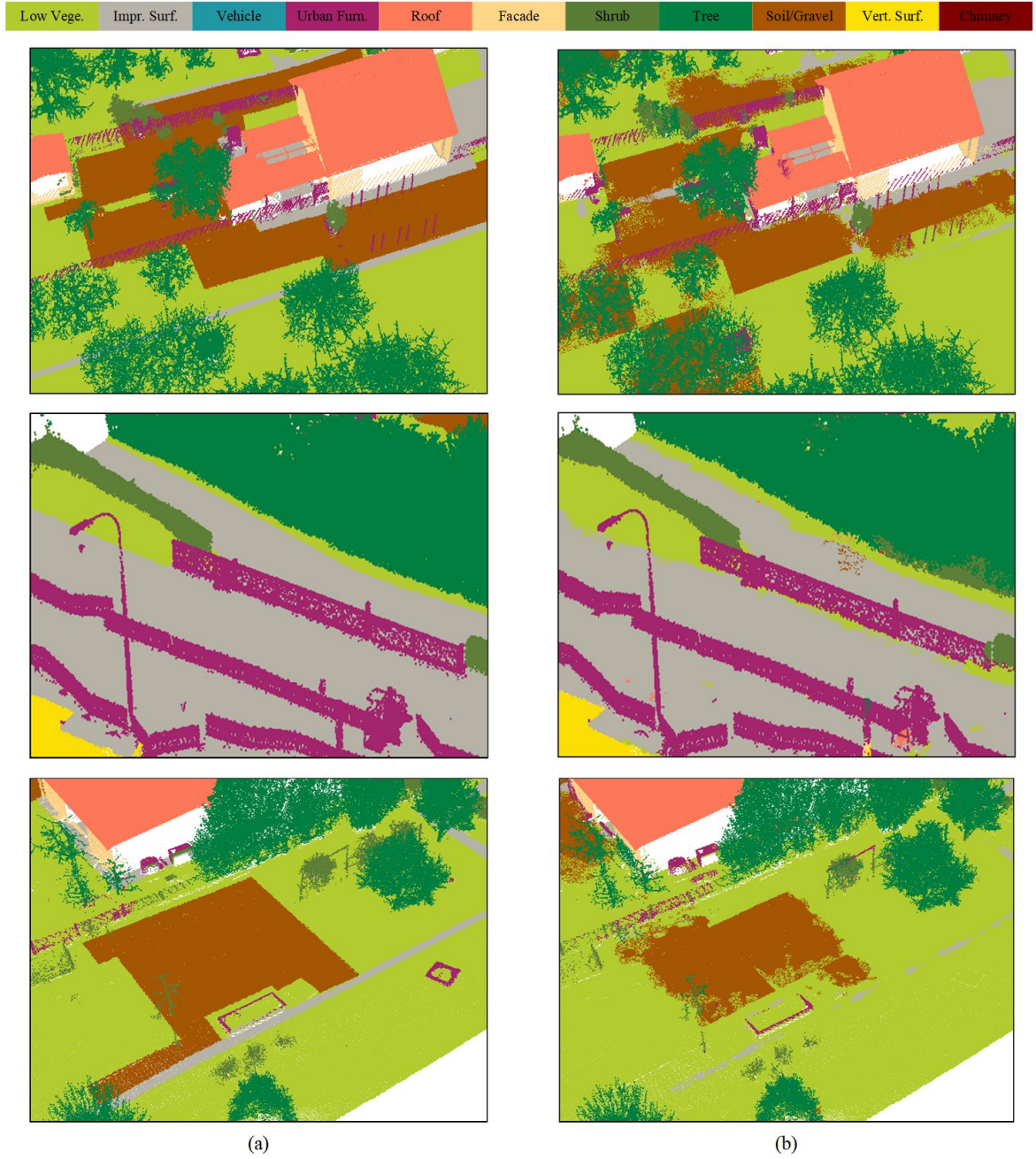


Figure B1: Detailed classification results using proposed SCE module on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

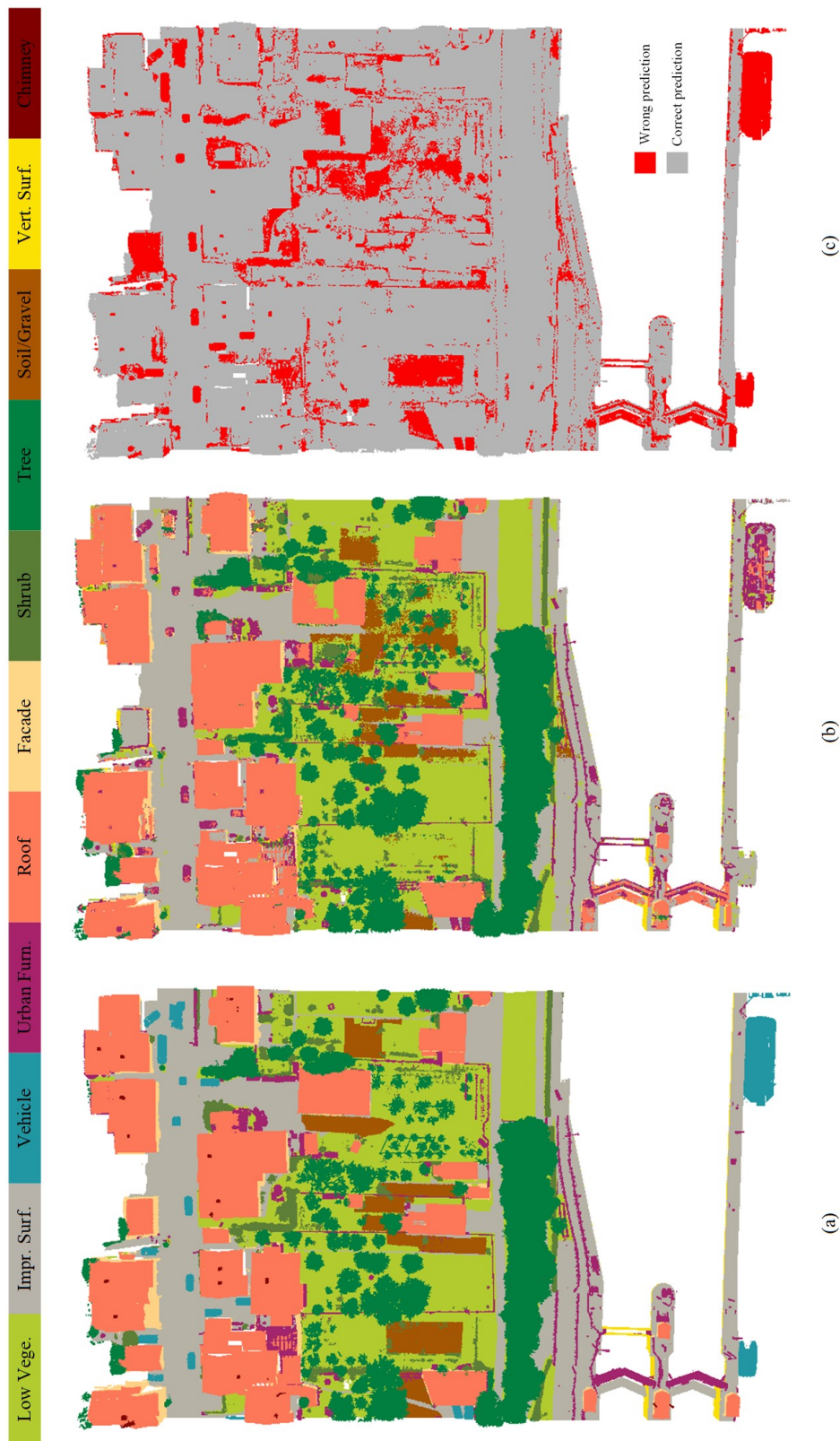


Figure B2: Classification results using proposed SCE module on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

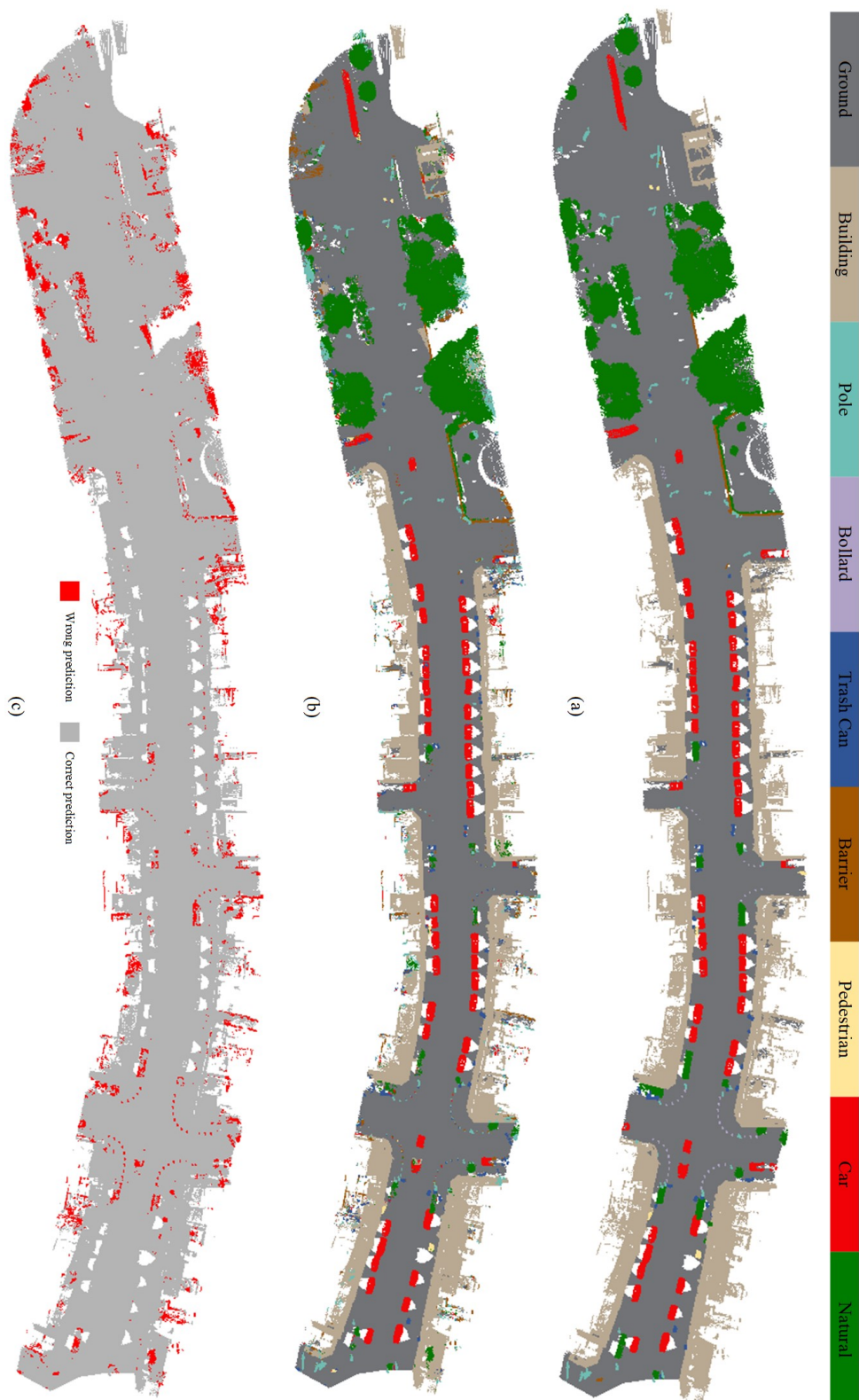


Figure B3: Classification results using proposed SCE module on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## C Supplementary Materials from the Contextual Joint Strategy

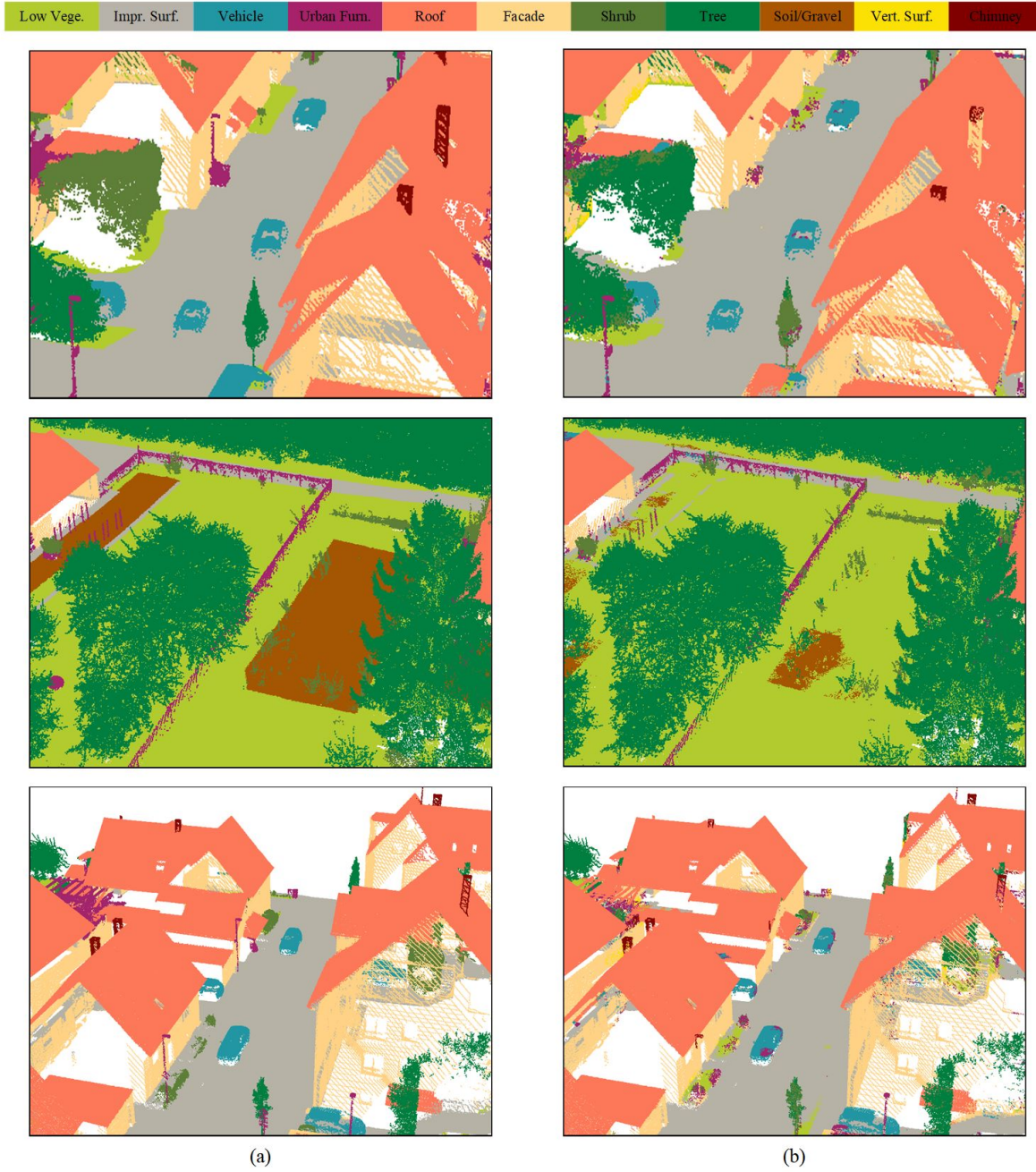


Figure C1: Detailed classification results using the contextual joint augmentation on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

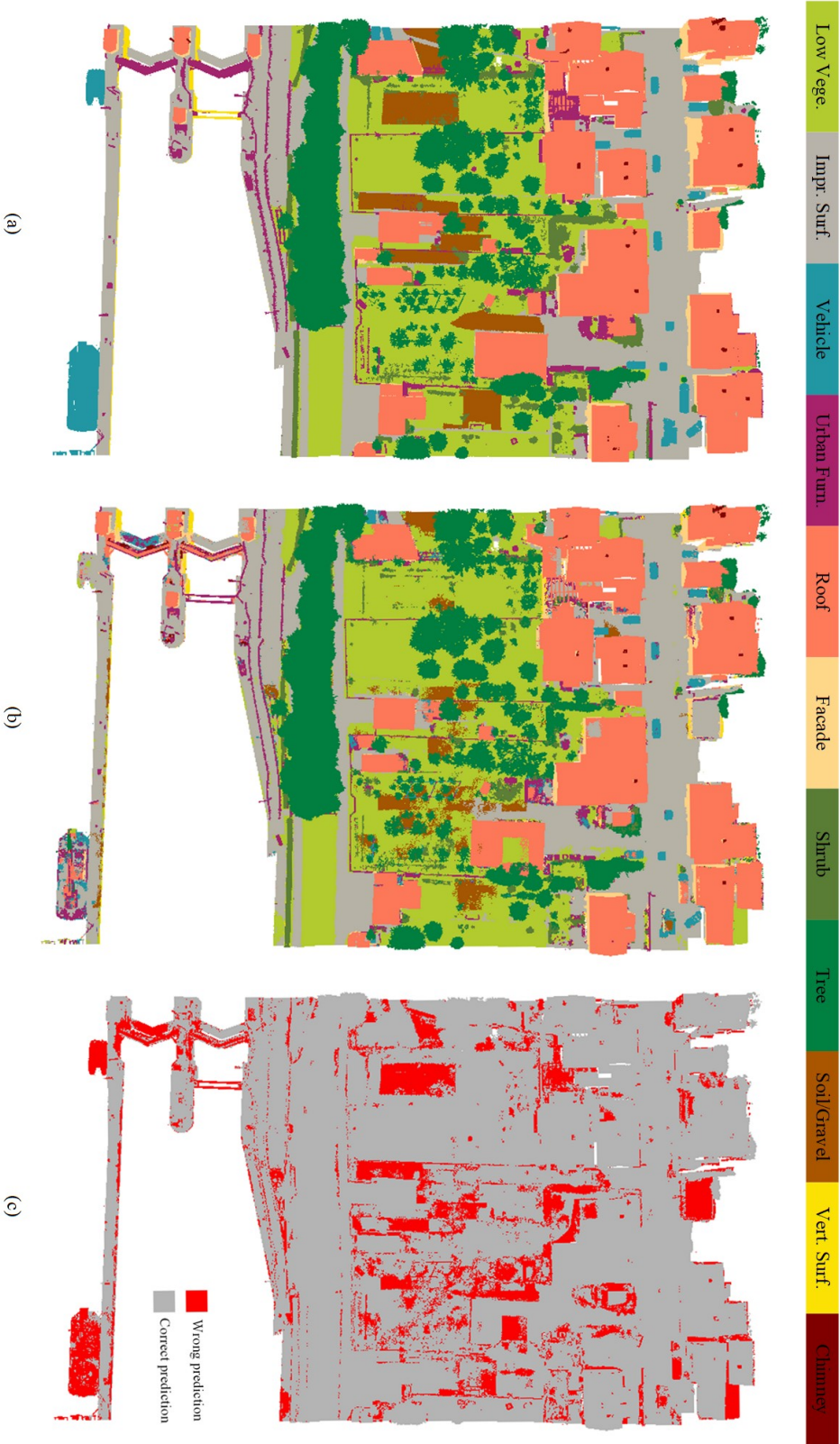


Figure C2: Classification results using the contextual joint augmentation on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

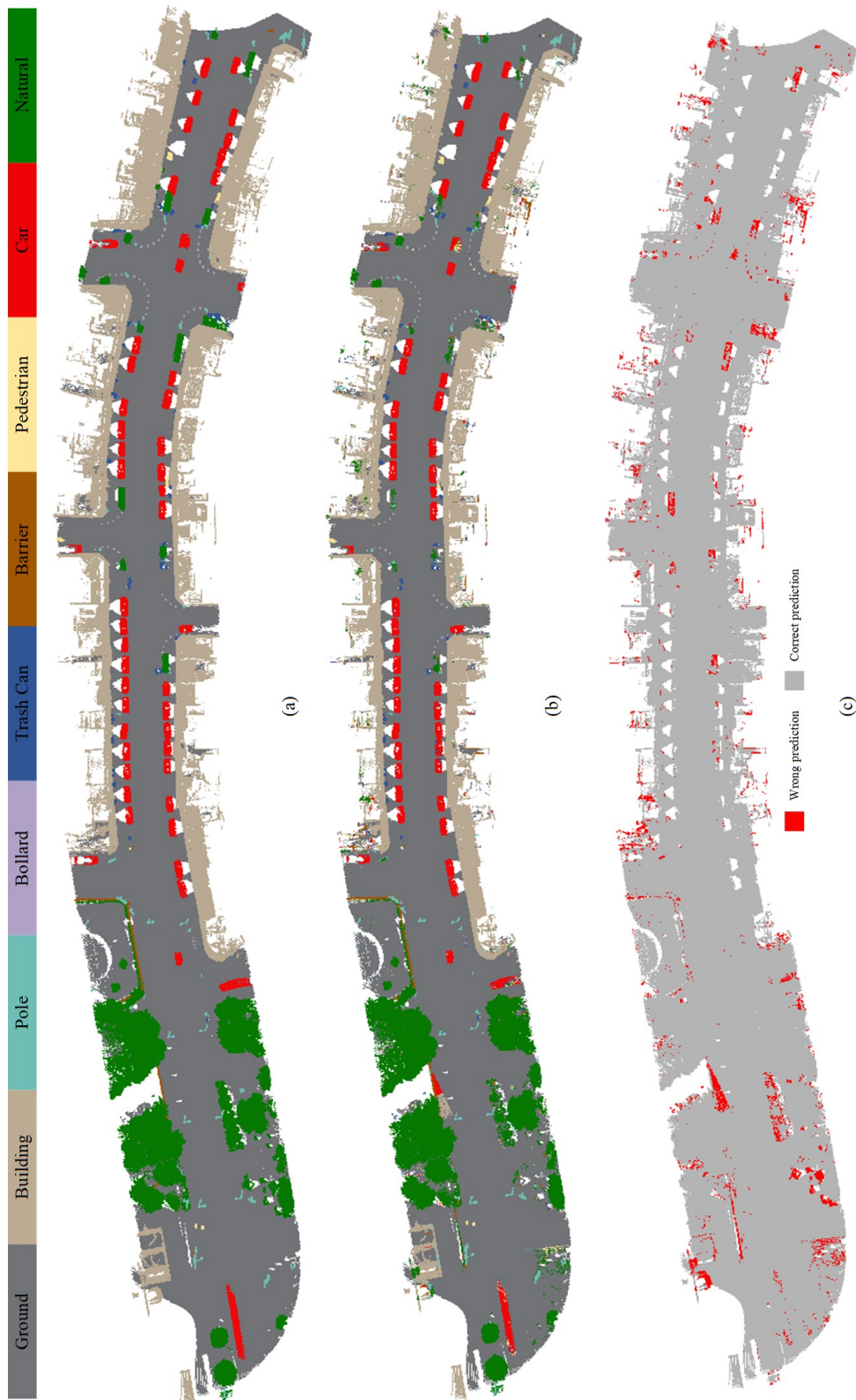


Figure C3: Classification results using the contextual joint augmentation on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

## D Supplementary Materials from the Target-aware Attention

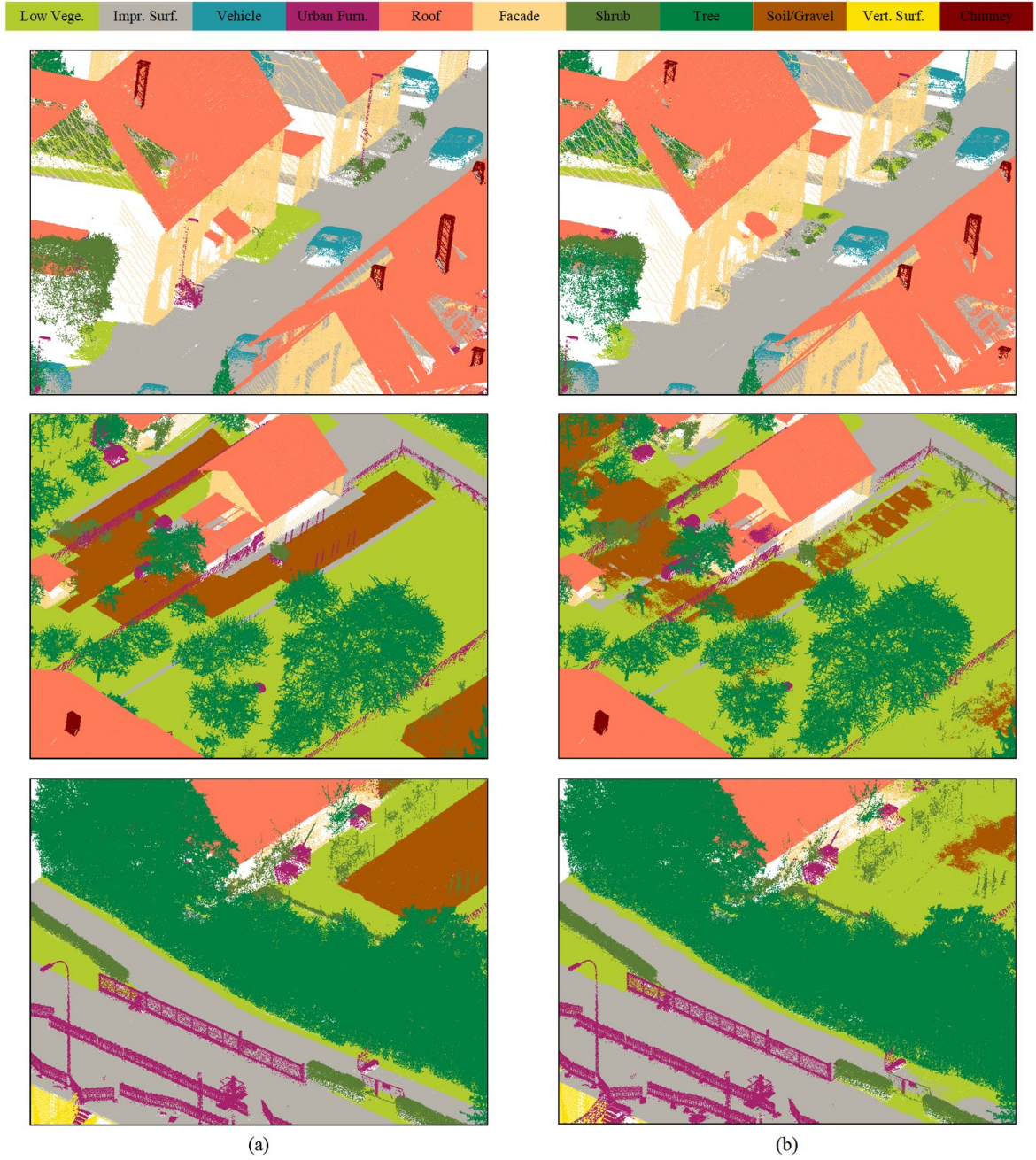


Figure D1: Detailed classification results of our TaaNet model on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification results of our method.

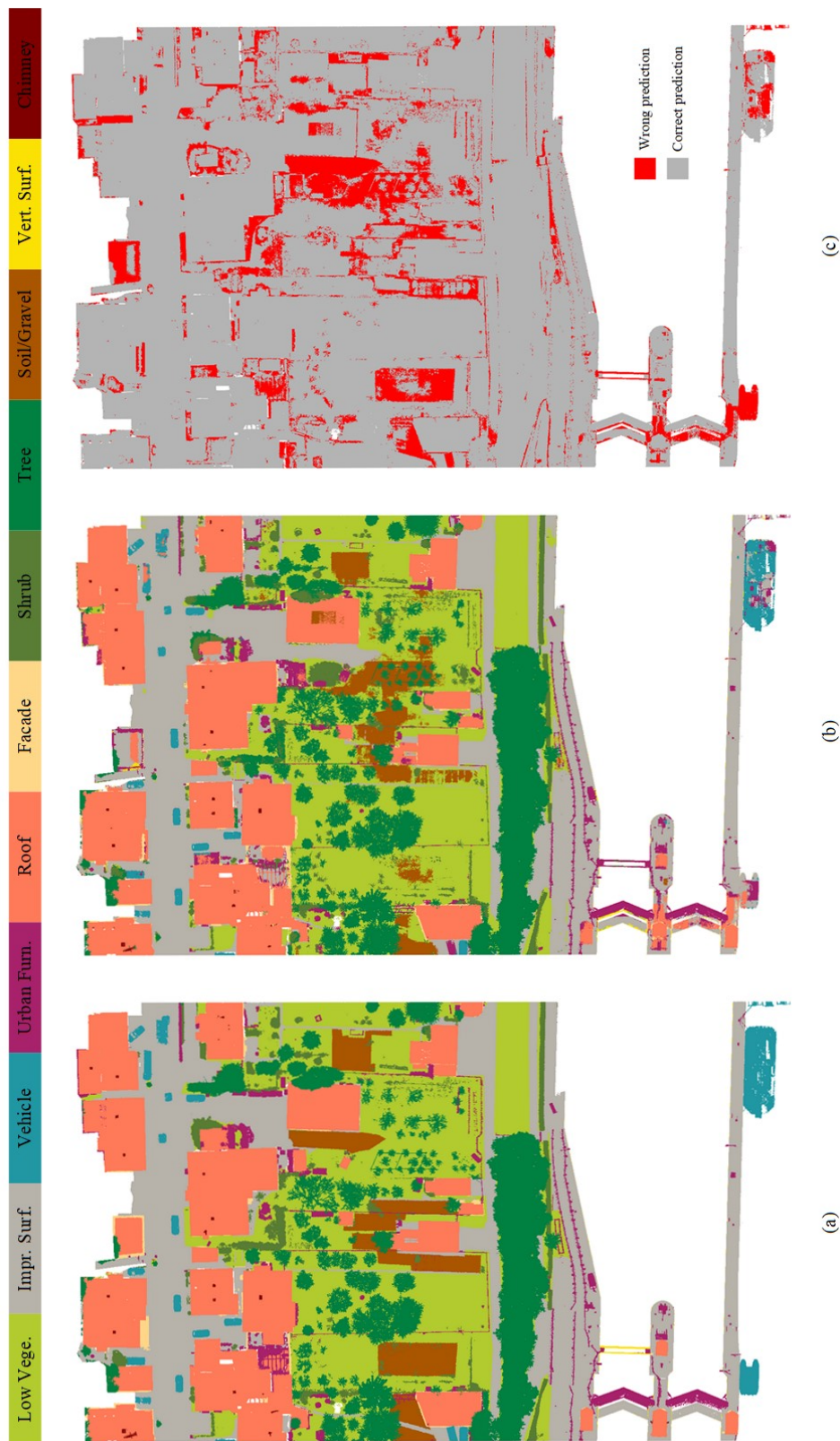


Figure D2: Classification results of our TaaNet model on the Hessigheim 3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

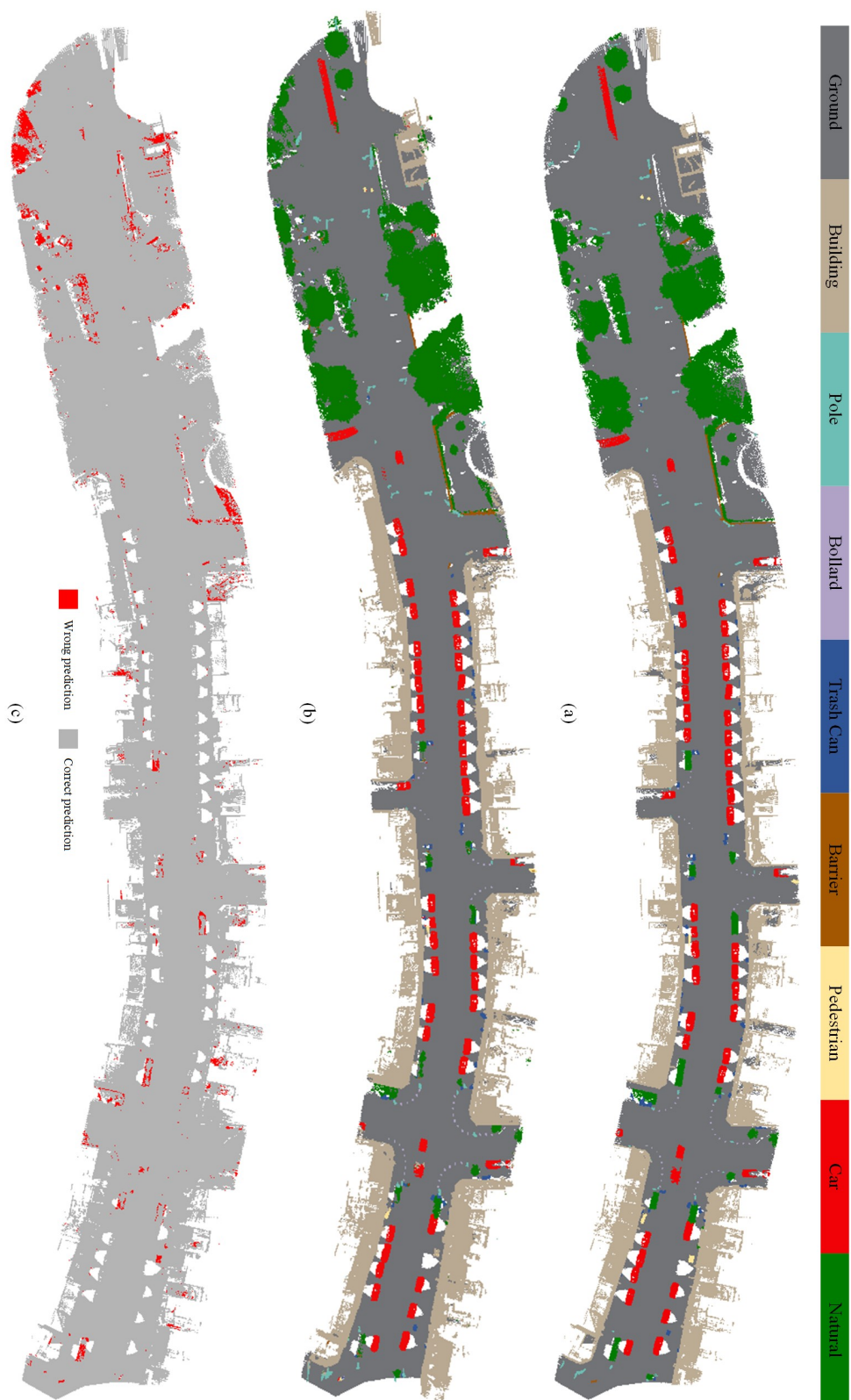


Figure D3: Classification results of our TaNet model on the Paris-Lille-3D dataset. (a) represents the ground truth. (b) presents the classification result using the proposed method. (c) presents the corresponding error map.

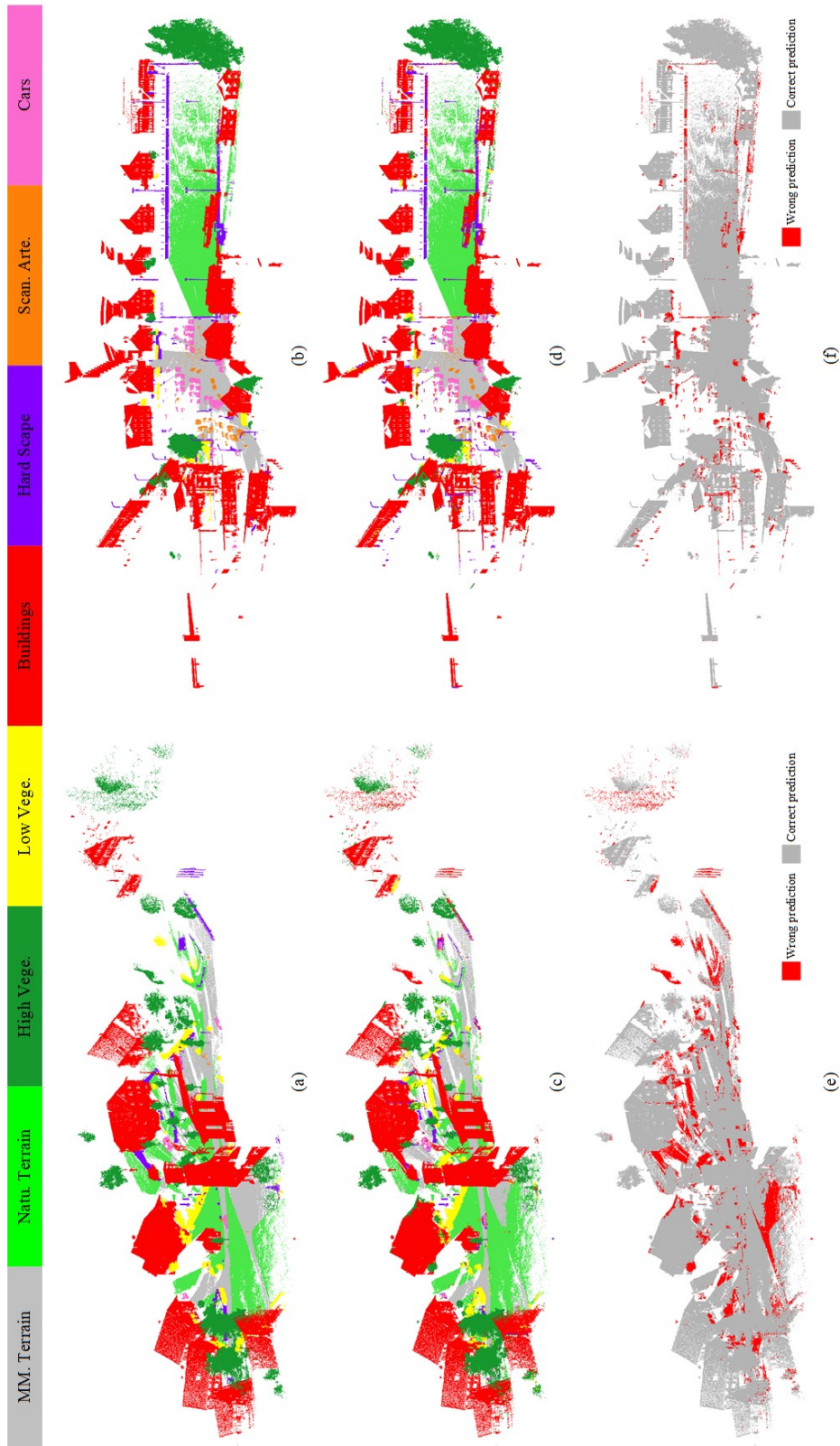


Figure D4: Classification results of our TaaNet model on the Semantic3D dataset. Only two selected scenes (bildstein5 and sg27\_9) are shown. (a) and (b) represent the ground truth. (c) and (d) present the classification results using the proposed method. (e) and (f) present the corresponding error maps.

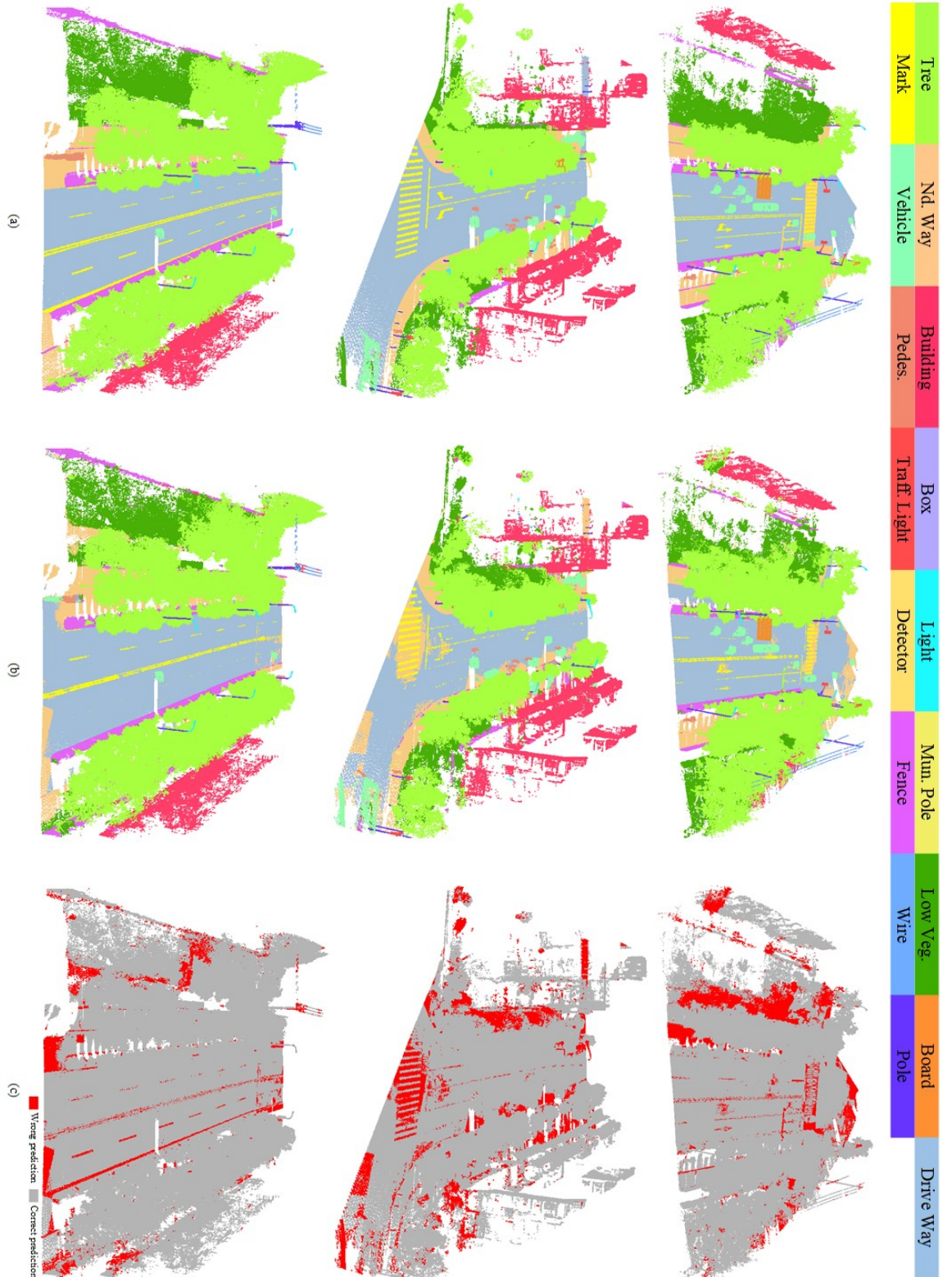


Figure D5: Classification results of our TaatNet model on MLS-S in the WHU-Urban3D dataset. Only three selected scenes are shown. Column (a) represents the ground truths. Column (b) presents the classification results using the proposed method. Column (c) presents the corresponding error maps.

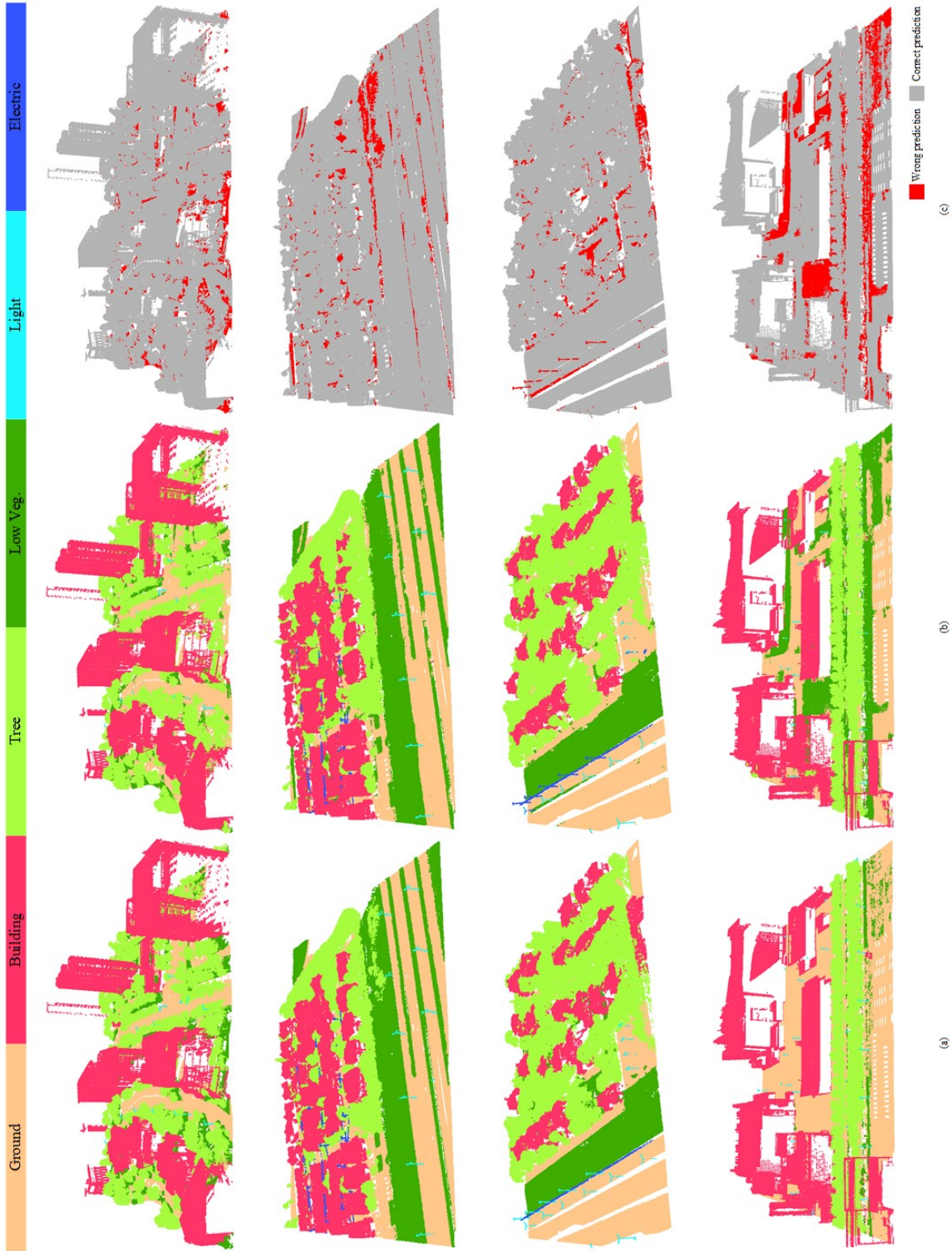


Figure D6: Classification results of our TaaNet model on ALS-S in the WHU-Urban3D dataset. Only four selected scenes are shown. Column (a) represents the ground truths. Column (b) presents the classification results using the proposed method. Column (c) presents the corresponding error maps.

To align with the predefined evaluation criteria for various datasets, we provide additional IoU-based results for a more comprehensive assessment in this section.

| Method        | IoU          |                |                |                |             |             |             |             |                 |                |                | OA          | mIoU        |
|---------------|--------------|----------------|----------------|----------------|-------------|-------------|-------------|-------------|-----------------|----------------|----------------|-------------|-------------|
|               | Low<br>Vege. | Impe.<br>Surf. | <u>Vehicle</u> | Urban<br>Furn. | Roof        | Facade      | Shrub       | Tree        | Soil/<br>Gravel | Vert.<br>Surf. | <u>Chimney</u> |             |             |
| PointNet++    | 79.0         | 80.1           | 32.5           | 28.0           | 85.7        | 55.7        | 33.2        | 90.1        | <b>38.5</b>     | 52.0           | 0.0            | 85.8        | 52.2        |
| MinkowskiNet  | 78.4         | 80.9           | 36.7           | 41.0           | 89.7        | 62.6        | 37.9        | 89.2        | 23.7            | 55.9           | 65.3           | 86.6        | 60.1        |
| DGCNN         | 74.6         | 81.4           | 67.6           | 39.5           | 90.4        | 64.1        | <b>46.0</b> | <b>92.6</b> | 31.7            | 61.4           | 34.2           | 86.6        | 62.1        |
| PACConv       | <b>79.2</b>  | 82.6           | 58.2           | 46.0           | 91.9        | 68.6        | 41.1        | 91.7        | 29.8            | <b>69.0</b>    | 58.2           | 88.1        | 65.1        |
| SPVConv       | 78.9         | 79.6           | 37.1           | 44.8           | 90.1        | 63.5        | 42.0        | 90.6        | 27.4            | 48.6           | 69.8           | 86.9        | 61.1        |
| TaaNet (Ours) | 78.6         | <b>82.9</b>    | <b>68.7</b>    | <b>49.5</b>    | <b>92.3</b> | <b>69.2</b> | 44.7        | 91.7        | 29.4            | 62.3           | <b>74.7</b>    | <b>88.2</b> | <b>67.6</b> |

Table D1: Comparison of the semantic segmentation results on the Hessigheim 3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in OA, IoU, and mean IoU are highlighted in bold text.

| Method        | IoU         |             |             |                |                      |             |                   |             |             | OA          | mIoU        |
|---------------|-------------|-------------|-------------|----------------|----------------------|-------------|-------------------|-------------|-------------|-------------|-------------|
|               | Ground      | Building    | <u>Pole</u> | <u>Bollard</u> | <u>Trash<br/>Can</u> | Barrier     | <u>Pedestrian</u> | Car         | Natural     |             |             |
| PointNet++    | 98.5        | 93.1        | 46.0        | 0.0            | 12.2                 | 6.4         | 3.9               | 89.0        | 60.6        | 95.9        | 45.5        |
| MinkowskiNet  | 98.4        | 98.0        | 71.1        | 72.4           | 73.1                 | 36.0        | <b>64.9</b>       | 97.1        | 79.7        | 98.4        | 76.7        |
| DGCNN         | 98.8        | 98.7        | 67.6        | 49.7           | 53.2                 | 34.8        | 60.8              | 96.7        | 76.5        | 98.4        | 70.7        |
| PACConv       | 98.8        | 97.9        | 74.1        | 49.1           | 67.5                 | 35.2        | 62.7              | 94.6        | 82.4        | 98.5        | 73.6        |
| SPVConv       | 98.6        | 98.4        | 76.8        | 77.8           | 80.3                 | 30.1        | 63.4              | 95.3        | 78.6        | 98.5        | 77.7        |
| TaaNet (Ours) | <b>98.9</b> | <b>98.9</b> | <b>84.8</b> | <b>78.6</b>    | <b>83.5</b>          | <b>61.3</b> | 62.7              | <b>97.9</b> | <b>85.5</b> | <b>99.0</b> | <b>83.6</b> |

Table D2: Comparison of the semantic segmentation results on the Paris-Lille-3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in OA, IoU, and mean IoU are highlighted in bold text.

| Method        | IoU            |                  |               |              |             |               |                |             | OA          | mIoU        |
|---------------|----------------|------------------|---------------|--------------|-------------|---------------|----------------|-------------|-------------|-------------|
|               | MM.<br>Terrain | Natu.<br>Terrain | High<br>Vege. | Low<br>Vege. | Buildings   | Hard<br>Scape | Scan.<br>Arte. | <u>Cars</u> |             |             |
| PointNet++    | 92.7           | <b>93.0</b>      | 69.6          | 39.7         | 89.3        | 21.1          | 31.0           | 60.4        | 90.5        | 62.1        |
| MinkowskiNet  | 91.5           | 64.3             | 75.4          | 48.4         | 87.4        | 26.9          | 56.1           | 67.2        | 88.9        | 64.7        |
| DGCNN         | 91.6           | 73.8             | 83.5          | 44.5         | 90.3        | 10.5          | 38.3           | 71.5        | 90.3        | 63.0        |
| PACConv       | 94.6           | 79.0             | 84.3          | 33.7         | 91.3        | 29.6          | 37.3           | 74.2        | 91.2        | 65.5        |
| SPVConv       | 92.5           | 64.1             | 69.9          | 48.6         | 87.7        | <b>31.6</b>   | 56.6           | 69.8        | 89.7        | 65.1        |
| TaaNet (Ours) | <b>96.8</b>    | 89.5             | <b>91.0</b>   | <b>58.1</b>  | <b>92.3</b> | 31.4          | <b>61.1</b>    | <b>82.0</b> | <b>93.7</b> | <b>75.3</b> |

Table D3: Comparison of the semantic segmentation results on the Semantic3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in OA, IoU, and mean IoU are highlighted in bold text.

| Method        | F1 Score    |             |             |             |             |                  |             |             |             |             |             |             |                     |                 |             |             |             | OA          | mIoU        |
|---------------|-------------|-------------|-------------|-------------|-------------|------------------|-------------|-------------|-------------|-------------|-------------|-------------|---------------------|-----------------|-------------|-------------|-------------|-------------|-------------|
|               | Tree        | Nd. Way     | Building    | <u>Box</u>  | Light       | <u>Mun. Pole</u> | Low Veg.    | Board       | Drive Way   | Mark        | Vehicle     | Pedes.      | <u>Traff. Light</u> | <u>Detector</u> | Fence       | Wire        | Pole        |             |             |
| PointNet++    | 88.4        | 45.9        | 67.9        | 0.1         | 35.3        | 0.0              | 23.7        | 0.9         | 87.9        | 53.9        | 76.3        | 26.2        | 12.5                | 0.0             | 44.2        | 0.0         | 29.9        | 83.0        | 34.9        |
| MinkowskiNet  | 90.8        | 36.7        | 64.9        | 35.1        | 32.2        | 5.0              | 27.1        | 23.9        | 85.5        | 49.3        | 82.3        | 60.8        | 41.3                | 26.4            | 49.7        | 19.7        | 60.0        | 83.5        | 46.5        |
| DGCNN         | 88.7        | 46.3        | 86.7        | 17.0        | 43.1        | 0.0              | 26.8        | 18.2        | <b>88.5</b> | 46.9        | 76.9        | 26.0        | 7.8                 | 0.0             | 48.0        | 0.0         | 44.6        | 84.3        | 39.1        |
| PAConv        | 90.2        | 45.2        | 88.0        | 45.1        | 39.4        | 0.0              | 27.8        | 29.4        | 88.1        | <b>54.5</b> | 88.3        | 68.9        | 8.7                 | 0.0             | <b>51.6</b> | 0.0         | 75.0        | <b>85.3</b> | 47.1        |
| SPVConv       | <b>91.3</b> | 40.7        | 64.0        | 33.4        | 41.5        | <b>9.0</b>       | 28.9        | 19.8        | 84.9        | 48.3        | 84.0        | 49.5        | 50.8                | 36.9            | 51.3        | 36.6        | 61.2        | 83.8        | 48.9        |
| TaaNet (Ours) | 91.0        | <b>47.1</b> | <b>90.7</b> | <b>75.8</b> | <b>50.2</b> | 7.7              | <b>38.2</b> | <b>64.2</b> | 86.2        | 46.8        | <b>93.9</b> | <b>74.8</b> | <b>74.1</b>         | <b>57.8</b>     | 49.7        | <b>54.4</b> | <b>83.6</b> | 85.2        | <b>63.9</b> |

Table D4: Comparison of the semantic segmentation results on MLS-S in the WHU-Urban3D dataset using our TaaNet and other baseline methods (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in OA, IoU, and mean IoU are highlighted in bold text.

| Method        | F1 Score    |             |             |             |              |             | OA          | mIoU        |
|---------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
|               | Ground      | Building    | Tree        | Low Veg.    | <u>Light</u> | Electric    |             |             |
| PointNet++    | 56.8        | 56.9        | 67.7        | 24.6        | 0.0          | 0.0         | 70.0        | 34.3        |
| MinkowskiNet  | 60.5        | 78.6        | 83.6        | 36.2        | 7.0          | 45.6        | 79.3        | 51.9        |
| DGCNN         | 63.7        | 78.2        | 84.3        | 39.5        | 7.1          | <b>85.3</b> | 81.1        | 59.7        |
| PAConv        | 64.6        | 80.3        | <b>86.1</b> | <b>47.7</b> | 22.0         | 84.1        | 82.7        | 64.1        |
| SPVConv       | 59.3        | 77.9        | 83.0        | 33.1        | 18.3         | 52.9        | 78.4        | 54.1        |
| TaaNet (Ours) | <b>68.1</b> | <b>86.3</b> | 85.1        | 46.4        | <b>49.2</b>  | 81.1        | <b>84.0</b> | <b>69.4</b> |

Table D5: Comparison of the semantic segmentation results on ALS-S in the WHU-Urban3D dataset using our TaaNet and other baseline methods. (values in %). Note that the rare categories in the dataset are marked in underlined text, and the highest values in OA, IoU, and mean IoU are highlighted in bold text.



# Acknowledgments

Looking back on this journey, I know that I have never walked it alone. It began with the trust and freedom placed in me by my supervisor, Prof. Dr.-Ing. Uwe Sörgel. I am deeply grateful to him for providing an outstanding research environment at the Institute for Photogrammetry and Geoinformatics (ifp). His guidance has shaped both this dissertation and my professional growth. I sincerely appreciate Prof. Dr. habil. Hans-Gerd Maas for serving as my co-referee and for his rigorous and constructive evaluation of this work. I am grateful to Prof. Dr.-Ing. Volker Schwieger for chairing the defense committee and for the time and effort he devoted to the evaluation process.

I have been fortunate to work with professional and warm-hearted people. My special thanks go to David Collmar and Wei Zhang, you have been steadfast companions throughout this journey. Your care and encouragement during times of mounting pressure helped me through the most challenging moments. I thank Dr.-Ing. Dong Lin for our technical discussions and exchanges, which broadened my research horizons and made even the most tedious debugging work intellectually engaging. My gratitude also goes to Dr.-Ing. Michael Kölle for his profound insights into the dataset and professional support, which laid a solid foundation for the experimental work of this study. I am equally grateful to Markus Englisch, whose excellent technical support enabled me to tackle complex computational tasks with confidence. I would like to thank all my colleagues at the ifp for the stimulating conversations during coffee breaks and in the corridors, which often sparked unexpected ideas.

My most heartfelt thanks are reserved for my family. To my parents, your unconditional love and support have given me the courage to explore the unknown. To my sister, the joy and companionship you bring are the gentlest light in my life. I am also grateful for the seemingly ordinary yet deeply precious moments along the way, whether an invigorating workout or an unexpected sunset. These fragments of everyday life, beyond the rigorous world of algorithms and data, have given me the warmth and vitality to keep moving forward.

Should I have forgotten anyone along the way, it is only because my gratitude overflows the text buffer. Please accept my sincere apologies and thanks.