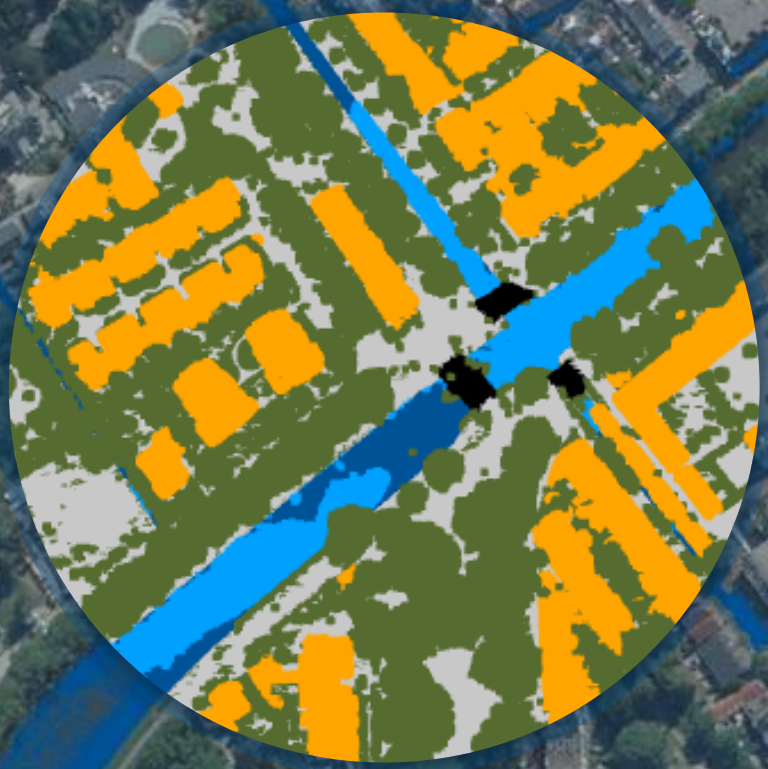
3D Urban Understanding from Point Clouds

Shenglan Du



TU Delft

3D URBAN UNDERSTANDING FROM POINT CLOUDS

3D URBAN UNDERSTANDING FROM POINT CLOUDS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus,
Prof. dr. ir. H. Bijl,
Chair of the Board for Doctorates
to be defended publicly on
Monday, 8 June 2026, 12:30

by

Shenglan DU

This dissertation has been approved by the promotors and the copromotor.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. J.E. Stoter	Delft University of Technology, promotor
Dr. J.F.P. Kooij	Delft University of Technology, promotor
Dr. L. Nan	Delft University of Technology, copromotor
Independent members:	
Prof. dr. ing. S. Nijhuis	Delft University of Technology
Prof. Dr. -Ing. N. Haala	University of Stuttgart, Germany
Prof. Dr. -Ing. N. Pfeifer	Vienna University of Technology, Austria
Dr. R.C. Lindenbergh	Delft University of Technology
Prof. Dr. -Ing. U. Pottgiesser	Delft University of Technology, reserve member

This thesis was financially supported by TU Delft AI Initiative.

Keywords: Point clouds, urban scene understanding, deep learning, semantic segmentation, urban tree instance segmentation, data-efficient segmentation

ISBN 978-94-6518-281-0

An electronic version of this dissertation is available at https://repository.tudelft.nl/.

CONTENTS

Summary ix

Samenvatting xi

Acronyms xiii

1 Introduction 1
- 1.1 Background and motivation . . . 1
- 1.2 Problem statement . . . 4
- 1.3 Research objectives and questions . . . 6
- 1.4 Thesis outline . . . 7

2 Fundamentals of point clouds and 3D urban understanding 9
- 2.1 Point cloud overview . . . 10
  - 2.1.1 Data acquisition techniques . . . 10
  - 2.1.2 Point cloud features . . . 11
- 2.2 3D semantic segmentation . . . 12
  - 2.2.1 Projection-based methods . . . 12
  - 2.2.2 Voxelization-based methods . . . 14
  - 2.2.3 Point-based methods . . . 14
  - 2.2.4 Summary . . . 18
- 2.3 3D instance segmentation . . . 18
  - 2.3.1 Proposal-based methods . . . 18
  - 2.3.2 Proposal-free methods . . . 19
  - 2.3.3 Object-specific instance segmentation . . . 20
  - 2.3.4 Summary . . . 21
- 2.4 Applications of segmented point clouds . . . 22

3 Local boundary-guided semantic segmentation 25
- 3.1 Introduction . . . 26
- 3.2 Related work . . . 28
  - 3.2.1 Graph-based segmentation . . . 28
  - 3.2.2 Boundary-aware refinement . . . 29
- 3.3 Method . . . 29
  - 3.3.1 Network architecture . . . 30
  - 3.3.2 Boundary-guided feature propagation . . . 31
  - 3.3.3 Network Supervision . . . 34

3.4 Experiments . . . 35
3.4.1 Datasets . . . 35
3.4.2 Implementation details . . . 36
3.4.3 Evaluation metrics . . . 37
3.4.4 Results of indoor scenes . . . 38
3.4.5 Results of outdoor scenes . . . 42
3.4.6 Ablation studies . . . 42
3.4.7 Complexity and convergence analysis . . . 47
3.4.8 Limitations . . . 48
3.5 Conclusion . . . 48

4 Global prototype expansion for semantic segmentation 51
4.1 Introduction . . . 52
4.2 Related work . . . 54
4.2.1 3D transformers and attention . . . 54
4.2.2 Prototype learning . . . 55
4.3 Method . . . 56
4.3.1 Prototype construction and updating . . . 56
4.3.2 Prototype expansion . . . 57
4.3.3 Analysis . . . 59
4.4 Experiments . . . 60
4.4.1 Evaluation setup . . . 60
4.4.2 Implementation details . . . 62
4.4.3 Results of indoor scenes . . . 62
4.4.4 Results of outdoor scenes . . . 66
4.4.5 Ablation studies . . . 69
4.4.6 Complexity and efficiency . . . 70
4.4.7 Limitations . . . 70
4.5 Conclusions . . . 71

5 Structure-aware tree instance segmentation 73
5.1 Introduction . . . 74
5.2 Related work . . . 77
5.2.1 Heuristic-based approaches . . . 77
5.2.2 Deep learning-based approaches . . . 77
5.3 Method . . . 79
5.3.1 Network Architecture . . . 80
5.3.2 Stem localization . . . 82
5.3.3 Tree point grouping . . . 84
5.4 Experiments . . . 85
5.4.1 Datasets . . . 85
5.4.2 Implementation details and hyperparameters . . . 86
5.4.3 Comparison and evaluation . . . 86
5.4.4 Results of urban forestry scenes . . . 87
5.4.5 Results of nature forestry scenes . . . 89
5.4.6 Ablation studies . . . 92

5.4.7 Limitations . . . 95
5.4.8 Potential applications . . . 95
5.5 Conclusions . . . 95

6 Confidence-based online learning for real-world point clouds 97
6.1 Introduction . . . 98
6.2 Related work . . . 100
6.3 Method . . . 101
6.3.1 Confidence measurement . . . 101
6.3.2 Online learning . . . 104
6.3.3 Class-balanced supervision . . . 105
6.4 Experiments . . . 106
6.4.1 Dataset . . . 106
6.4.2 Implementation details and hyperparameters . . . 106
6.4.3 Quantitative results . . . 107
6.4.4 Qualitative results . . . 109
6.4.5 Label refinement on training data . . . 110
6.4.6 Point density analysis . . . 112
6.4.7 Building footprint comparison . . . 112
6.4.8 Limitations and applications . . . 114
6.5 Conclusions . . . 114

7 Conclusions 117
7.1 Contributions and key findings . . . 118
7.2 Future work . . . 122

Acknowledgements 145

Curriculum Vitæ 147

List of Publications 149

SUMMARY

Automated analysis and interpretation of 3D urban environments from laser-scanned point clouds has emerged as a critical research area with broad applications in urban planning, land administration, autonomous driving, and navigation. Despite remarkable progress in this field, researchers face two key challenges: (i) the comparatively slower advancement of methodologies for 3D point cloud analysis compared to 2D image-based techniques, and (ii) the difficulty of scaling these methods to large and complex real-world urban environments. This thesis addresses both aspects by exploring methodological innovations in 3D point cloud processing and investigating their applicability to large-scale urban settings, with an overall aim of supporting more robust and reliable interpretation of 3D urban scenes.

The first emphasis of this thesis is the development of fundamental deep learning methodologies for 3D point cloud processing, specifically for the task of 3D semantic segmentation, which aims to parse the urban scenes into semantically meaningful parts. We have proposed two core technical contributions to effectively address the limitations in existing research. First, from a local spatial perspective, a boundary-guided refinement approach has been introduced to better preserve fine-grained local boundary details and enhance semantic boundary delineation; Second, zooming to a global perspective, a prototype expansion module has been proposed to efficiently encode global semantic context as a set of prototype embeddings, thereby mitigating the restricted receptive fields of conventional approaches and enhancing overall segmentation performance. Together, the two technical contributions improve the network's feature learning capacity from both local and global perspectives, leading to more accurate, context-aware, and robust segmentation of complex urban environments.

Another focus of this thesis is the extension and scaling of 3D point cloud learning methodologies to large-scale urban environments, with a particular emphasis on specific urban objects such as trees. A novel structure-aware framework has been introduced to explicitly leverage tree structural priors for robust segmentation of individual tree instances. Compared to existing methods, this approach significantly improves segmentation accuracy, especially in challenging urban areas with complex and heterogeneous forestry structures.

Last, this thesis investigates the practical deployment of contemporary deep learning architectures on real-world point cloud datasets, which are often affected by noise, outliers, and annotation errors. To address these challenges, we have proposed a confidence-based online learning framework that prioritizes high-confidence point

samples during training while iteratively refining the labels of low-confidence points. This approach effectively improves both the robustness of semantic segmentation and the overall annotation quality of real-world point cloud datasets.

In summary, this thesis establishes a comprehensive framework for understanding 3D urban environments from point clouds by integrating methodological innovations with practical deployment strategies. Beyond advancing the state of the art in segmentation and interpretation, the proposed approaches lay the groundwork for scalable, reliable, and adaptive solutions in real-world urban applications, providing a solid foundation for future research in the intelligent processing, analysis, and interpretation of large-scale 3D point cloud data.

SAMENVATTING

Geautomatiseerde analyse en interpretatie van 3D-stedelijke omgevingen op basis van lasergescande puntenwolken is uitgegroeid tot een cruciaal onderzoeksgebied met brede toepassingen in stedenbouw, landbeheer, autonoom rijden en navigatie. Ondanks de opmerkelijke vooruitgang op dit gebied, staan onderzoekers voor twee belangrijke uitdagingen: (i) de relatief trage ontwikkeling van methodologieën voor 3D-puntenwolkanalyse in vergelijking met 2D-beeldgebaseerde technieken, en (ii) de moeilijkheid om deze methoden op te schalen naar grote en complexe stedelijke omgevingen in de praktijk. Deze thesis behandelt beide aspecten door methodologische innovaties in 3D-puntenwolkverwerking te onderzoeken en de toepasbaarheid ervan op grootschalige stedelijke omgevingen te bestuderen, met als overkoepelend doel een robuustere en betrouwbaardere interpretatie van 3D-stedelijke scènes te ondersteunen.

De eerste focus van dit proefschrift ligt op de ontwikkeling van fundamentele deep learning-methodologieën voor de verwerking van 3D-puntenwolken, specifiek voor de taak van 3D-semantische segmentatie, die tot doel heeft stedelijke scènes op te delen in semantisch betekenisvolle onderdelen. We hebben twee belangrijke technische bijdragen voorgesteld om de beperkingen in bestaand onderzoek effectief aan te pakken. Ten eerste is vanuit een lokaal ruimtelijk perspectief een grensgestuurde verfijningsaanpak geïntroduceerd om fijnmazige lokale grensdetails beter te behouden en de semantische grensafbakening te verbeteren; Ten tweede, vanuit een mondiaal perspectief, is een prototype-uitbreidingsmodule voorgesteld om de globale semantische context efficiënt te coderen als een set prototype-embeddings. Hierdoor worden de beperkte receptieve velden van conventionele benaderingen verminderd en de algehele segmentatieprestaties verbeterd. Samen verbeteren deze twee technische bijdragen het vermogen van het netwerk om kenmerken te leren vanuit zowel lokaal als globaal perspectief, wat leidt tot een nauwkeurigere, contextbewuste en robuuste segmentatie van complexe stedelijke omgevingen.

Een ander aandachtspunt van dit proefschrift is de uitbreiding en opschaling van 3D-puntwolk-leermethoden naar grootschalige stedelijke omgevingen, met bijzondere nadruk op specifieke stedelijke objecten zoals bomen. Er is een nieuw, structuurbewust raamwerk geïntroduceerd om expliciet gebruik te maken van de structurele voorkennis van bomen voor robuuste segmentatie van individuele boomsoorten. Vergeleken met bestaande methoden verbetert deze aanpak de segmentatienauwkeurigheid aanzienlijk, met name in uitdagende stedelijke gebieden met complexe en heterogene bosstructuren.

Tot slot onderzoekt dit proefschrift de praktische toepassing van hedendaagse deep learning-architecturen op realistische puntwolk-datasets, die vaak beïnvloed worden door ruis, uitschieters en annotatiefouten. Om deze uitdagingen aan te pakken, hebben we een op betrouwbaarheid gebaseerd online leerframework voorgesteld dat prioriteit geeft aan puntmonsters met een hoge betrouwbaarheid tijdens de opleiding, terwijl de labels van punten met een lage betrouwbaarheid iteratief worden verfijnd. Deze aanpak verbetert effectief zowel de robuustheid van de semantische segmentatie als de algehele annotatiekwaliteit van realistische puntwolk-datasets.

Samenvattend biedt dit proefschrift een alomvattend raamwerk voor het begrijpen van 3D-stedelijke omgevingen aan de hand van puntenwolken, door methodologische innovaties te integreren met praktische implementatiestrategieën. Naast het bevorderen van de stand van de techniek op het gebied van segmentatie en interpretatie, leggen de voorgestelde benaderingen de basis voor schaalbare, betrouwbare en adaptieve oplossingen in realistische stedelijke toepassingen. Dit biedt een solide fundament voor toekomstig onderzoek naar de intelligente verwerking, analyse en interpretatie van grootschalige 3D-puntenwolkdata.

Acronyms

AHN Actueel Hoogtebestand Nederland

AI Artificial Intelligence

ALS Airborne Laser Scanning

AP Average Precision

BAG Basisregistratie Adressen en Gebouwen

BIM Building Information Modeling

CHM Canopy Height Model

CNN Convolutional Neural Network

CRF Conditional Random Field

DNN Deep Neural Network

DSM Digital Surface Model

GAN Generative Adversarial Network

GFP Guided Feature Propagation

GT Ground Truth

IoU Intersection over Union

kNN k-Nearest Neighbor

LiDAR Light Detection and Ranging

Mask R-CNN Mask Region-based Convolutional Neural Network

mIoU mean Intersection over Union

MLP Multi-Layer Perceptron

MLS Mobile Laser Scanning

MRF Markov Random Field

MVS Multi-View Stereo

MSE Mean Squared Error

MST Minimum Spanning Tree

NDVI Normalized Difference Vegetation Index

NLP Natural Language Processing

OA Overall Accuracy

PE Prototype Expansion

PR Precision-Recall

ReLU Rectified Linear Unit

SIFT Scale-Invariant Feature Transform

SfM Structure from Motion

SFP Standard Feature Propagation

SPG SuperPoint Graph

S3DIS Stanford Large-Scale 3D Indoor Spaces

TLS Terrestrial Laser Scanning

TP Throughput

1

INTRODUCTION

1.1. BACKGROUND AND MOTIVATION

We live in a three-dimensional world composed of diverse urban entities, such as buildings, lampposts, roads, and vegetation. With the accelerated expansion of cities and the ongoing urbanization process, there is an ever-increasing demand from both academia and industry for advanced methodologies to understand and model the complex urban environment from rich-sourced spatial information (Biljecki *et al.*, 2015; Bouzas *et al.*, 2020). In particular, extensive research has been conducted on 3D urban scene understanding, aiming to develop computational frameworks for interpreting and reconstructing urban landscapes (Geiger *et al.*, 2011).

Understanding the urban scenes requires addressing two fundamental questions: (1) What are the objects in the scene? (2) Where are the objects in the scene? Answering these questions is key to the success of various applications in urban studies and geospatial information science. For instance, segmenting urban scenes into semantically meaningful objects facilitates the generation of high-resolution 3D city models (Stoter *et al.*, 2014), which play a critical role in urban digitalization, land administration, and city management. Meanwhile, precise object detection and localization within urban environments serve as foundational tasks in positioning (Hsu *et al.*, 2015), robotics navigation (Crespo *et al.*, 2020), and autonomous driving (Yurtsever *et al.*, 2020). Moreover, accurate urban scene interpretation provides valuable input for environmental analysis and urban simulation (García-Sánchez *et al.*, 2018), contributing to sustainable urban planning and ecological benefits for society.

Recent advancements in 3D data acquisition technologies, such as Light Detection and Ranging (LiDAR), have greatly encouraged the collection and utilization of 3D geospatial information for urban scene analysis. LiDAR systems capture 3D object surfaces as a set of discrete points, namely point clouds, within Euclidean space. Compared to 2D imagery, LiDAR point clouds inherently provide precise 3D measurements by preserving the raw geometries of objects in 3D, making them

1

a preferred data representation for urban environments (Landrieu *et al.*, 2017). However, unlike images embodied with regular grid structures, LiDAR point clouds exhibit sparsity, lack of order, and irregular spatial distribution. These characteristics present significant challenges for 3D urban scene understanding from point clouds, as conventional 2D image processing techniques, such as Convolutional Neural Networks (CNNs) (Krizhevsky *et al.*, 2012), may not be directly applicable or effective for analyzing 3D point cloud data.

(a) Urban scene rendered in meshes[1] (b) LiDAR point clouds with semantics

Figure 1.1: An example of 3D urban scene (a) represented with point clouds (b). We use the following colors to render distinct urban semantics: *building*, *vegetation*, *ground*, and *water*.

Early approaches to urban scene interpretation from point clouds primarily relied on manual annotation and visual inspection of the obtained urban data, which are labor-intensive and costly. With the emergence of Artificial Intelligence (AI), specifically deep learning technologies, classifying and segmenting urban point clouds has become increasingly automated over the past few decades. Traditional machine learning methods typically employ a two-step pipeline: first, hand-crafted point features are extracted, followed by the training of a supervised classifier to assign point labels (Niemeyer *et al.*, 2014, 2016; Thomas *et al.*, 2018; Weinmann *et al.*, 2015). Nevertheless, these approaches often require considerable domain expertise and extensive feature engineering to ensure high-quality representations. On the contrary, recent deep learning techniques enable direct processing of raw point cloud data and generate strong feature maps by learning to represent objects as a nested hierarchy of concepts, discarding the human efforts to design domain-specific features (Goodfellow, 2016).

The paradigm shift from conventional machine learning to more recent deep learning has significantly enhanced the efficiency and accuracy of numerous tasks such as natural language processing (Young *et al.*, 2018), image analysis (Krizhevsky *et al.*, 2012), and 3D computer vision (Qi, Su, *et al.*, 2017). Specifically, a large number of deep learning-based frameworks have been developed for 3D urban scene

[1]Image is sourced from Google Earth: https://earth.google.com/.

interpretation from point cloud data, including projection-based methods (H. Guo *et al.*, 2016; Qi *et al.*, 2016; H. Su *et al.*, 2015), voxelization-based methods (Choy *et al.*, 2019; Maturana & Scherer, 2015; Riegler *et al.*, 2017), and point-based methods (Lai *et al.*, 2022; H. Lin *et al.*, 2023; Qi, Su, *et al.*, 2017; Qi, Yi, *et al.*, 2017; Qian *et al.*, 2022; Thomas *et al.*, 2019; X. Wu *et al.*, 2024; H. Zhao *et al.*, 2021).

Despite the remarkable progress in this field, deep learning-based 3D urban scene analysis from LiDAR point clouds still faces critical challenges. One major challenge is the comparatively slower development of deep learning-based methods for point cloud processing relative to image-based approaches. Another challenge lies in how to effectively scale these frameworks to real-world urban environments with the increased complexity, larger spatial extent, and greater data uncertainties. To bridge these gaps, this thesis advances the core methodologies of 3D point cloud learning and investigates their application to large-scale, real-world urban settings, both of which are critical for automated object interpretation and precise spatial localization in urban environments.

Primarily, this thesis focuses on advancing deep learning methodologies for point cloud processing, with an emphasis on the task of 3D semantic segmentation, which aims to assign a class label (e.g., building, vegetation, water, and road) to each point to achieve a structured interpretation of the urban layouts. Contemporary 3D point cloud learning methods, including MLP-based (H. Lin *et al.*, 2023; Qi, Yi, *et al.*, 2017; Qian *et al.*, 2022), convolution-based (Y. Li *et al.*, 2018; Thomas *et al.*, 2019), and transformer-based architectures (Lai *et al.*, 2022; X. Wu *et al.*, 2022; H. Zhao *et al.*, 2021), have demonstrated promising performance in various urban scene interpretation tasks. However, most approaches exhibit two key limitations: First, from a local spatial perspective, segmentation accuracy near object boundaries is often degraded due to the loss of fine-grained geometric details; Second, from a global perspective, limited receptive fields greatly hinder the ability to capture long-range semantic dependencies and contextual relationships among urban objects. To cope with these challenges, this thesis proposes two core technical contributions: (i) a boundary-guided local refinement mechanism designed to preserve and enhance semantic boundary delineation; (ii) a Prototype Expansion method that efficiently encodes global semantic contextual information to improve urban scene interpretation. Collectively, these technical innovations improve the representational capacity of the network, which contributes to more accurate, context-aware, and robust segmentation of complex urban scenes.

In addition to fundamental technical advancements, this thesis also investigates methodologies for scaling 3D point cloud learning frameworks to large-scale and highly complex real-world urban environments. First, it targets a representative urban object category, i.e., tree, proposing a structure-aware deep learning framework for large-scale tree segmentation at the instance level. Our selection of trees is motivated by their ecological and aesthetic importance in urban landscapes, as well as the structural complexity they introduce to urban scene analysis. Existing heuristic-based (Lee *et al.*, 2010; J. Wang *et al.*, 2018; Yun *et al.*, 2021) and learning-based (Hakula *et al.*, 2023; Henrich *et al.*, 2024; T. Jiang, Wang, *et al.*, 2023)

methods for individual tree segmentation often struggle in complex forestry scenes, particularly when confronted with dense canopies, occlusions, and varying tree geometries. We propose to overcome this limitation by explicitly leveraging structural priors, such as tree stems, to enable robust detection and delineation of individual tree instances under challenging conditions. Furthermore, in this thesis, we explore the practical deployment of contemporary deep learning architectures on real-world datasets, such as the Dutch airborne LiDAR dataset (AHN, 2025). To address the semantic segmentation challenges related to data uncertainties and annotation errors in real-world scenes, we propose a confidence-aware online learning framework that prioritizes high-confidence point samples during training while iteratively refining the labels of low-confidence points. Our approach effectively enhances both the segmentation robustness and the annotation quality of real-world point clouds.

By integrating these technical innovations as well as their practical deployments, this thesis establishes a robust framework for 3D urban scene understanding from point clouds. Our proposed methods are designed to be adaptable across different data types, varying scene scales, and inherent data uncertainties, thereby improving both the generalizability and applicability of deep learning models for urban scene analysis. In Section 1.2, we formally state our research problems. In Section 1.3, we provide a detailed articulation of the research objectives and questions.

1.2. PROBLEM STATEMENT

This thesis investigates the automated interpretation and understanding of 3D urban scenes from LiDAR point clouds. Focusing on both fundamental methodology advancements and practical deployment, four key problems have been identified to overcome the limitations of existing research.

Problem I. *Addressing semantic segmentation from a local perspective: Concurrent point cloud learning methods, particularly point-based CNNs, often exhibit inconsistencies in scene parsing and struggle with precise local boundary delineation.*

Semantic segmentation of LiDAR point clouds is a crucial task that benefits various downstream applications, including urban modeling, environmental analysis, and autonomous driving. Inspired by the success of Deep Neural Networks (DNNs) in 2D imagery processing, numerous 3D point cloud learning networks have been introduced (Y. Li *et al.*, 2018; Qi, Yi, *et al.*, 2017; Thomas *et al.*, 2019, 2024; M. Xu, Ding, *et al.*, 2021). While promising, these networks severely suffer from poor object boundary delineation due to the loss of fine-grained local details. Although a few works have attempted to alleviate boundary segmentation errors by enforcing additional smoothness constraints (L.-C. Chen *et al.*, 2017; Zheng *et al.*, 2015) or incorporating object boundaries as priors (Gong *et al.*, 2021; Z. Hu *et al.*, 2020), they do not explicitly tackle the fundamental issue of information loss, which greatly bottlenecks the network's capacity to learn strong feature maps at high resolutions, thereby leading to suboptimal segmentation performance in local boundary regions.

Problem II. *Addressing semantic segmentation from a global perspective: There is a shortage of effective global-level point cloud learning methods to facilitate consistent and efficient segmentation across entire scenes.*

The introduction of PointNet and PointNet++(Qi, Su, *et al.*, 2017; Qi, Yi, *et al.*, 2017) marked a significant breakthrough in point cloud processing. Following them, numerous point-based deep learning approaches have been developed, with a primary focus on designing local feature aggregation functions (Y. Li *et al.*, 2018; H. Lin *et al.*, 2023; X. Ma *et al.*, 2022; Thomas *et al.*, 2019). Despite their strong performances on various point cloud interpretation tasks, they share a fundamental limitation: The receptive fields are constrained since the feature learning operators are typically applied within local neighborhood regions. Theoretically, it is possible to perform global operators, such as the attention mechanism, over the entire scene. However, in practice, applying such mechanisms to large-scale point clouds is computationally prohibitive due to the high volume of input points. Striking a balance between computational efficiency and effective global contextual reasoning remains a critical open challenge in the field.

Problem III. *Addressing real-world tree segmentation at the instance level: Trees contribute significant ecological value, but also introduce considerable challenges to urban scene analysis. Existing instance segmentation methods remain inadequate for large-scale applications, particularly in handling variations in tree size and shape.*

3D instance segmentation represents a finer-grained interpretation of urban scenes by partitioning scenes into distinct object instances. Among common urban objects, buildings and trees play a vital role as they constitute a substantial portion of the built environment. While buildings can often be extracted and reconstructed using footprint data (Peters *et al.*, 2023), modeling trees is more challenging due to their complex and irregular structures. Moreover, accurate mapping of individual urban trees brings significant value to urban environmental simulation, ecosystem analysis, and forestry management (Maltamo *et al.*, 2014). Traditional tree instance segmentation methods rely on heuristics and domain-specific knowledge (Hakula *et al.*, 2023; Lee *et al.*, 2010; J. Wang *et al.*, 2018). With the advances of deep learning, researchers have explored learning-based methods for tree instance segmentation (Henrich *et al.*, 2024; T. Jiang, Wang, *et al.*, 2023; H. Luo *et al.*, 2021; P. Wang *et al.*, 2023). However, most existing methods struggle to accurately identify individual trees in complex urban forests, where trees exhibit dense overlap, occlusions, and varying geometries. There is still a pressing need for developing advanced learning-based methods to robustly segment 3D tree instances at large scales, especially in intricate forestry environments.

Problem IV. *Towards data-efficient learning for real-world urban scene interpretation: Effective semantic segmentation methods tailored for real-world point clouds, which contain outliers, data uncertainties, and annotation errors, are notably lacking.*

The vast majority of existing methods for 3D point cloud learning are fully

supervised, assuming the availability of dense point-wise annotations. However, acquiring such high-quality annotations is both costly and time-consuming. Real-world datasets, such as Dutch LiDAR point cloud (AHN, 2025), inevitably come with data noises, outliers, and labeling errors, making them challenging to analyze using well-established fully supervised methods. Several studies have explored data-efficient learning strategies for point cloud understanding, such as weakly supervised learning (Yao *et al.*, 2024), unsupervised pre-training (Xie *et al.*, 2020), and knowledge distillation (Y. Liu *et al.*, 2024). However, the effectiveness and applicability of these approaches on large-scale, real-world datasets such as the Dutch point cloud remain largely unexplored.

1.3. RESEARCH OBJECTIVES AND QUESTIONS

The main research objective of this thesis is to develop and implement an automatic and robust framework for the semantic and instance-level interpretation of 3D urban scenes from LiDAR point clouds. Our proposed framework aims to enhance semantic segmentation by improving local consistency and global contextual reasoning. Additionally, we investigate fine-grained instance segmentation, specifically targeting challenging urban forestry trees at large scales. Last, it explores data-efficient learning strategies for real-world datasets such as airborne LiDAR point clouds. To this end, the following research questions have been defined for each research problem identified in Section 1.2.

Problem I. *Addressing semantic segmentation from a local perspective: Concurrent point cloud learning methods, particularly point-based CNNs, often exhibit inconsistencies in scene parsing and struggle with precise local boundary delineation.*

Research questions:

1. *What are the underlying reasons of suboptimal boundary delineation in existing point cloud learning approaches?*
2. *How can boundary priors be effectively integrated to reduce segmentation errors and enhance local-level consistency?*

This thesis will address these questions by developing an automatic 3D semantic segmentation algorithm to explicitly mitigate segmentation inconsistencies and ambiguities near object boundaries.

Problem II. *Addressing semantic segmentation from a global perspective: There is a shortage of effective global-level point cloud learning methods to facilitate consistent and efficient segmentation across entire scenes.*

Research questions:

1. *What factors limit the capacity of deep neural networks to perform effective*

global-level analysis on point clouds?

2. *How can we design a feature operating module that leverages global contextual knowledge to enhance semantic segmentation, while minimizing computational and memory costs?*

This thesis will answer these questions by developing a network module to perform efficient global analysis for point cloud learning and urban scene understanding.

Problem III. *Addressing real-world tree segmentation at the instance level: Trees contribute significant ecological value, but also introduce considerable challenges to urban scene analysis. Existing instance segmentation methods remain inadequate for large-scale applications, particularly in handling variations in tree size and shape.*

Research questions:

1. *Which shape characteristics distinguish trees from other types of urban objects and can be exploited for instance segmentation tasks?*
2. *How to effectively address tree overlap, occlusion, and geometric variations in challenging urban forestry areas?*

This thesis will address the questions by developing a robust deep learning-based approach for automated 3D instance segmentation of trees in large-scale urban and nature forestry scenes.

Problem IV. *Towards data-efficient learning for real-world urban scene interpretation: Effective semantic segmentation methods tailored for real-world point clouds, which contain outliers, data uncertainties, and annotation errors, are notably lacking.*

Research questions:

1. *How can data uncertainties be measured and leveraged to improve semantic understanding of point clouds in real-world environments?*
2. *How well does the confidence-based approach perform on real-world airborne point clouds, and what are its potential applications?*

This thesis will answer these questions by developing a data-efficient learning strategy for the interpretation of real-world urban point clouds, accounting for data outliers, uncertainties, and annotation defects.

1.4. THESIS OUTLINE

This thesis consists of seven chapters. Chapter 1 introduces the topic, outlines key open challenges, and defines the corresponding research objectives. Subsequent chapters are structured to address each specific research objective:

- Chapter 2 introduces the fundamentals of point cloud data and background knowledge on 3D urban scene understanding, with a focus on recent advancements in deep learning for scene segmentation at the semantic and instance levels.
- Chapter 3 presents a boundary-guided approach to enhance local-level semantic segmentation near object boundaries. It incorporates boundary priors to guide the feature propagation process, thereby preserving local structure details and reducing boundary errors. Its effectiveness is rigorously evaluated on indoor and outdoor urban scenes.
- Chapter 4 explores global contextual reasoning for urban scene analysis. It introduces a novel approach that encodes global scene-level knowledge into class prototype embeddings, establishing feature-wise associations between individual points and class descriptors through a Prototype Expansion mechanism, which is flexible and easy to plug into existing networks. Evaluation across various datasets has demonstrated the effectiveness of the proposed method.
- Chapter 5 focuses on fine-grained 3D segmentation at the instance level, specifically targeting urban trees at large scales. It is driven by the inherent complexities and significant ecological importance of tree segmentation. The chapter introduces a structure-aware learning-based method that jointly segments crowns and stems, enhancing the accuracy of individual tree identification. This approach is extensively evaluated on urban and nature forestry datasets, demonstrating its robustness across diverse 3D forestry environments.
- Chapter 6 investigates 3D segmentation of real-world data, proposing a data-efficient strategy designed to cope with the impact of outliers, uncertainties, and annotation inconsistencies. By incorporating data uncertainty measures, this strategy refines semantic segmentation by leveraging partial point cloud samples for learning. Experiments on Dutch point cloud datasets (AHN, 2025) have validated its effectiveness and applicability in real-world urban scenes.
- Chapter 7 summarizes the research performed by this thesis, including key conclusions, critical reflections, and future perspectives. Additionally, it provides recommendations for further research to extend this study and enhance the automated analysis, interpretation, and monitoring of 3D urban environments.

2

FUNDAMENTALS OF POINT CLOUDS AND 3D URBAN UNDERSTANDING

This chapter presents fundamental principles to understand urban environments from 3D point clouds. It provides an overview of point cloud acquisition methods, data types, and inherent properties. It also examines recent advancements in automated techniques for parsing, analyzing, and interpreting urban scenes from point clouds, with a particular emphasis on two key 3D perception tasks: semantic segmentation and instance segmentation. Besides, this chapter discusses the importance of 3D semantic and instance-level information in various urban socio-ecological applications.

2.1. POINT CLOUD OVERVIEW

A point cloud is a discrete set of points in the Euclidean space, with each point defined by three Cartesian coordinates (x, y, z). The points naturally capture the external surface and geometric structure of an object. Due to their simplicity, unified representation, and capacity to accurately model 3D geometry, point clouds have become a vital data type in surveying, with widespread applications across computer vision and computer graphics, including urban reconstruction, virtual reality, and autonomous driving. However, there are also challenges in using point cloud data. First, the points are sparse and irregularly distributed in the 3D space, making it difficult to analyze them for 3D urban interpretation tasks. Additionally, it often requires massive amounts of data (e.g., millions) to fully capture an urban scene, which can pose significant scalability challenges to existing solutions due to the high data volume and processing complexity.

2.1.1. DATA ACQUISITION TECHNIQUES

Point clouds can be derived from 2D sources, such as images, using photogrammetry techniques. This is done by combining multiple overlapping images from disparate viewpoints and simultaneously optimizing the camera parameters and 3D key point coordinates to reconstruct scene geometry. Structure from Motion (SfM) and Multi-View Stereo (MVS) are among the most widely adopted close-range photogrammetry methods (Schonberger & Frahm, 2016; Snavely *et al.*, 2006), which employ robust feature detectors such as Scale-Invariant Feature Transform (SIFT) (Lowe, 1999) to automate the generation of high-resolution point clouds. With the availability of various SfM and MVS systems, they have gained significant prominence in topographical analysis (James & Robson, 2012), landscape modeling (Smith & Vericat, 2015), and other geoscience applications (Westoby *et al.*, 2012).

In contrast to photogrammetry, the more straightforward approach to acquiring point clouds is through LiDAR. A LiDAR system is implemented by emitting pulsed or modulated laser beams to target an object's surface and measuring precise distances based on the time of travel for the reflected laser signal (Taylor, 2019). The output of laser scanning is high-resolution point clouds that are ideally suited for leveling and surveying purposes. LiDAR point clouds can be collected through various methodologies, including Airborne Laser Scanning (ALS), Mobile Laser Scanning (MLS), and Terrestrial Laser Scanning (TLS) (Shan & Toth, 2018):

- **ALS:** The LiDAR system is deployed on an aerial platform to acquire large-scale topographic data, which is particularly effective for extensive land surveying.
- **MLS:** The LiDAR scanner is mounted on a moving vehicle to dynamically capture 3D spatial data, which is commonly applied in autonomous driving and road mapping.
- **TLS:** The LiDAR system is positioned on a stationary platform to perform high-precision scanning, which is widely used in architectural documentation,

construction monitoring, and archaeological studies.

(a) ALS (b) MLS (c) TLS

Figure 2.1: Three laser scanning techniques to obtain LiDAR data.[1]

Figure 2.1 illustrates the ALS, MLS, and TLS systems for acquiring LiDAR point clouds. With the rapid advancement of these laser scanning technologies, the collection and utilization of 3D point cloud data have expanded significantly. In particular, numerous publicly available datasets have recently been released with dense semantic annotations, such as S3DIS (Armeni *et al.*, 2016), ScanNet (Dai *et al.*, 2017), ISPRS (Rottensteiner *et al.*, 2012), PL3D (Roynard *et al.*, 2018), Semantic3D (Hackel *et al.*, 2017), Toronto3D (W. Tan *et al.*, 2020), DALES (Varney *et al.*, 2020), and SensatUrban (Q. Hu *et al.*, 2021). These datasets have played a vital role in advancing deep learning-based point cloud analysis for urban 3D understanding, facilitating the development of novel methods for 3D segmentation at both the scene and instance levels.

2.1.2. POINT CLOUD FEATURES

LiDAR point cloud data stores precise 3D spatial measurements along with additional point-wise spectral attributes, including RGB color, intensity, and number of return. These features are crucial for the semantic interpretation of urban 3D scenes. For example, LiDAR return intensity could be combined with aerial image intensity to classify the input points into distinct categories such as buildings, trees, roads, and grass (Lodha *et al.*, 2006).

Besides spectral features, researchers have made numerous efforts in designing handcrafted features, which refer to data properties manually engineered by humans based on domain knowledge priors or specific algorithms, to encode statistical or contextual properties of the points. Most existing studies focused on extracting discriminative shape descriptors characterized by the local neighborhood of individual points. One strategy is to exploit point feature histograms (Rusu

[1]Images are sourced from online.
ALS: https://commons.wikimedia.org/wiki/File:Airborne_Laserscan-DSC_0089w.jpg
MLS: https://www.mdpi.com/2072-4292/10/4/492
TLS: https://forschungsinfrastruktur.bmbwf.gv.at/en/fi/3d-terrestrial-laser-scanner-riegl-vz-4000_3715

et al., 2008) or point orientation histograms (Tombari *et al.*, 2010) to describe local geometric structures. Another strategy is to compute the covariance matrix from the 3D coordinates of neighboring points to analyze local surface variations. Using the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the covariance matrix, one can derive eigen-based features to quantitatively describe the local geometrical properties such as linearity, planarity, and sphericity (Weinmann *et al.*, 2015; West *et al.*, 2004). Table 2.1 summarizes commonly used eigenfeatures for point cloud analysis.

	Formulation
linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
sphericity	λ_3/λ_1
omnivariance	$\sqrt[3]{\lambda_1\lambda_2\lambda_3}$
eigen-entropy	$-\sum_{i=1}^{3}\lambda_i log\lambda_i$

Table 2.1: Commonly used eigenfeatures for 3D point cloud analysis (Weinmann *et al.*, 2015). $\lambda_1, \lambda_2, \lambda_3$ denote the eigenvalues of the covariance matrix derived from the local neighborhood of a given point and $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

Handcrafted features are often integrated with LiDAR spectral attributes for 3D scene analysis. These features can be used as input to supervised classifiers, such as Support Vector Machine (J. Zhang *et al.*, 2013), Random Forest (Niemeyer *et al.*, 2013), Gaussian Mixture Model (Lalonde *et al.*, 2005), or Bayesian Discriminant Classifier (Khoshelham *et al.*, 2013), to achieve point-wise semantic classification. Nevertheless, for a specific urban scene, it is not trivial to find the optimal combination of features (Qi, Su, *et al.*, 2017) due to variations in scene complexity and feature relevance.

2.2. 3D SEMANTIC SEGMENTATION

Recent advancements in deep learning have greatly revolutionized 3D semantic segmentation from point clouds. Deep learning directly learns robust feature representations from raw input, eliminating the need for manually designed and selected features (see Section 2.1.2). Existing deep learning approaches for 3D semantic segmentation can be categorized into projection-based, discretization-based, and point-based methods (Y. Guo *et al.*, 2020).

2.2.1. PROJECTION-BASED METHODS

While standard deep learning networks, e.g., CNNs, have demonstrated remarkable success in 2D image recognition, their application on 3D point clouds remains challenging due to the irregular and sparse distribution of points in 3D space.

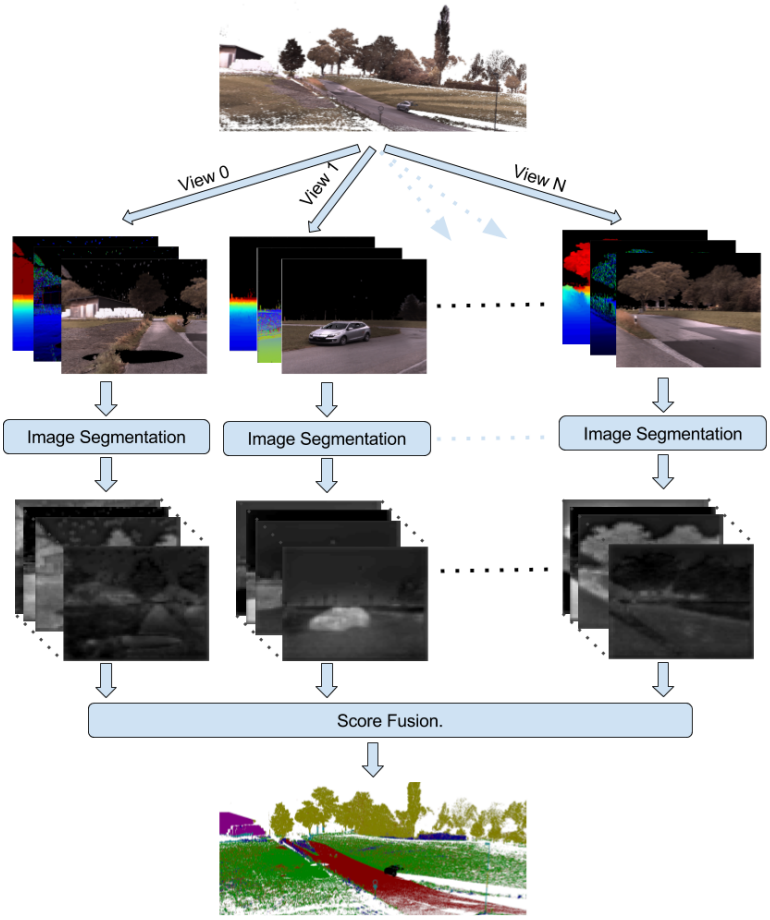


Figure 2.2: Deep projective 3D semantic segmentation (Lawin *et al.*, 2017).

To address this issue, projection-based methods first project 3D points onto 2D planes from multiple viewpoints and then use CNNs for 3D semantic recognition. The Multi-View Convolutional Neural Network, introduced by H. Su *et al.* (2015), constructs a multi-view representation of 3D input from virtual cameras and aggregates information from different perspectives to generate a compact and discriminative 3D feature descriptor. Similar strategies have been employed by H. Guo *et al.* (2016), Qi *et al.* (2016), Z. Yang and Wang (2019), and T. Yu *et al.* (2018) for 3D object recognition. Furthermore, projection-based approaches have also been adopted for 3D semantic segmentation and urban scene analysis (Lawin *et al.*, 2017). As depicted in Figure 2.2, semantic classification is performed on projected pseudo-images and subsequently mapped back to the original point clouds.

2.2.2. VOXELIZATION-BASED METHODS

Another approach for enabling standard CNNs to process irregular point clouds is to voxelize the points within a 3D volumetric grid. VoxNet (Maturana & Scherer, 2015) and ShapeNet (Z. Wu *et al.*, 2015) are among the earliest studies that extend convolutional networks to operate on voxelized point clouds. However, volumetric representations often incur high computational costs. To address this, Riegler *et al.* (2017) proposed an octree-based 3D representation, reducing runtime and memory storage by leveraging hierarchical octrees to store sparse point clouds. Graham *et al.* (2018) proposed submanifold sparse convolutional networks, which optimize the processing of voxelized data using standard U-Net architectures (Çiçek *et al.*, 2016). Choy *et al.* (2019) developed the Minkowski Engine, an auto-differentiation framework for sparse tensors, incorporating generalized sparse convolutions for 3D voxel-based learning.

Nevertheless, even with improved computational efficiency, voxelization can introduce discretization artefacts and information loss, potentially leading to suboptimal performance in downstream tasks.

2.2.3. POINT-BASED METHODS

Contrary to projection-based and voxelization-based methods, point-based methods directly process raw point clouds with deep neural networks, eliminating the need for intermediate transformations. Over the past decade, these methods have gained significant attention and have become the predominant approach for 3D urban scene analysis. Based on the design of point-wise feature learning mechanisms, point-based methods can be categorized into three main categories: point Multi-Layer Perceptron (MLP) networks, point convolutional networks, and point transformer networks.

POINT MLP NETWORKS

An MLP is a fundamental computational unit in deep neural networks, comprising fully connected neurons with nonlinear activation functions (Cybenko, 1989). PointNet (Qi, Su, *et al.*, 2017), a seminal research in point cloud learning, employs a sequence of MLP layers to directly process raw 3D points without requiring intermediate data transformations. These MLP layers are shared across all points to encode point-wise features, followed by a global max-pooling layer that aggregates features into a permutation-invariant and robust global descriptor (Figure 2.3).

The original PointNet fails to capture local contextual relationships and fine-grained structural details in 3D scenes. To alleviate this issue, the follow-up PointNet++ (Qi, Yi, *et al.*, 2017) employs a hierarchical design to recursively apply PointNet over a nested partitioning of the point set. By combining features from multi-scale local neighborhoods and aggregating local contexts hierarchically, PointNet++ enhances its capability to model fine structural details. Inspired by PointNet and PointNet++, H. Zhao *et al.* (2019) proposed PointWeb, which learns

pairwise interactions among points by densely connecting neighboring points into a web-like structure. RandLaNet (Q. Hu *et al.*, 2020), on the other hand, improves the efficiency of processing large-scale point clouds by employing a fast point sampling strategy.

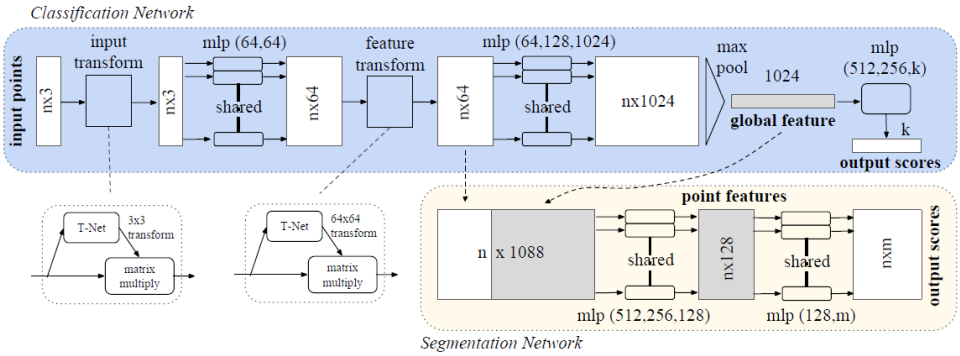


Figure 2.3: PointNet architecture (Qi, Su, *et al.*, 2017).

Since 2019, point MLP networks have been largely surpassed by more advanced architectures, such as point convolutional networks and transformers. Very recently, however, MLP-based networks were revisited with improved network design. Notably, incorporating geometric surface priors (Ran *et al.*, 2022) or enhancing local regional geometry learning (X. Ma *et al.*, 2022) has led to substantial performance gains in PointNet++. Furthermore, the studies of PointNext (Qian *et al.*, 2022) and PointMetabase (H. Lin *et al.*, 2023) have also shown that the classical PointNet++ can be significantly enhanced through advanced model scaling and network optimization techniques, enabling it to outperform many state-of-the-art networks such as Point Transformer (H. Zhao *et al.*, 2021). Figure 2.4 illustrates the architectural differences between PointNet++ and PointNext.

POINT CONVOLUTIONAL NETWORKS

Point-based convolution methods aim to define explicit convolutional operators that extend naturally to 3D space. Compared to MLP, point-based convolutions offer a more structured parameter-sharing mechanism, which enhances computational efficiency and accelerates network convergence.

One of the earliest convolution studies, PointCNN (Y. Li *et al.*, 2018), introduces an X-transformation that reorders and weights point features to ensure they are suitable for processing by a 3D convolutional operator. Several subsequent studies have been dedicated to parameterized kernel functions to extend convolutions over the full continuous vector space (S. Wang *et al.*, 2018; Y. Xu *et al.*, 2018). A representative method in this category, KPConv (Thomas *et al.*, 2019), defines convolutional kernels as a set of evenly distributed spherical 3D points, allowing pseudo-kernel features

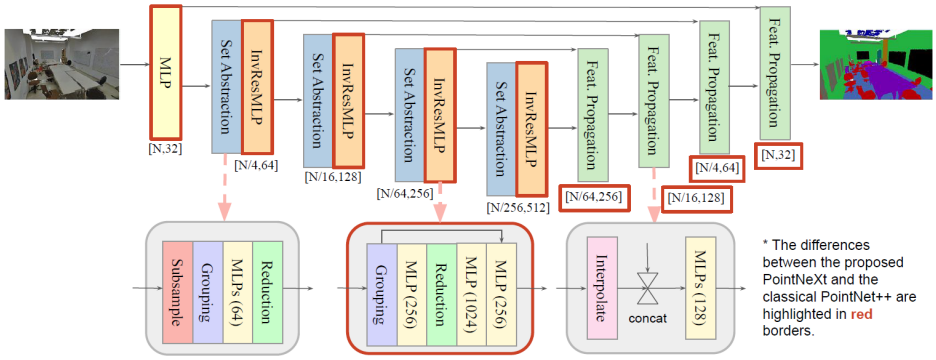


Figure 2.4: PointNext architecture (Qian *et al.*, 2022).

to be generated and processed using regular convolution operations (Figure 2.5). The flexibility to determine the number and the location of kernels also enables KPConv to perform deformable convolutions that learn to adapt kernel positions to local geometric structures. Following KPConv, PAConv (M. Xu, Ding, *et al.*, 2021) dynamically learns convolutional weights based on point position-aware embeddings, while KPConvX (Thomas *et al.*, 2024) further enhances the model adaptability by incorporating a kernel attention mechanism to scale convolutional weights dynamically.

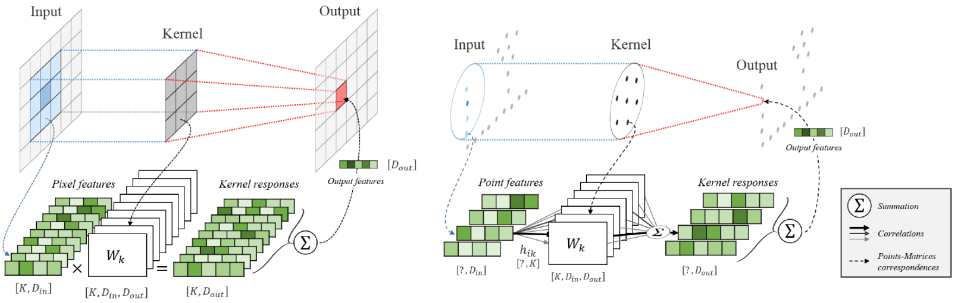


Figure 2.5: Extending 2D convolution (left) to 3D Kernel point convolution (right) (Thomas *et al.*, 2019).

POINT TRANSFORMER NETWORKS

Initially designed for Natural Language Processing (NLP) tasks (Vaswani *et al.*, 2017), transformer-based architectures have demonstrated great power in capturing long-range contextual dependencies among objects. This capability has enabled

transformers to surpass classical CNNs in various 2D image understanding tasks (Dosovitskiy, 2021; Kirillov *et al.*, 2023; Z. Liu *et al.*, 2021, 2022).

Furthermore, transformers are particularly well-suited for processing point clouds. The reason is that the self-attention mechanism, the core component of transformer networks, is by nature a set operator, which well respects the permutation and cardinality invariance of the input points. The first transformer network specifically designed for point cloud analysis was introduced by H. Zhao *et al.* (2021), as illustrated in Figure 2.6. Point Transformer implements self-attention within local k-Nearest Neighbor (kNN) regions, where each point attends to its neighbors to aggregate contextual features. Meanwhile, positional encoding, an essential component in NLP transformers, is integrated into the local attention mechanism to incorporate geometric information. Around the same time, PCT (M.-H. Guo *et al.*, 2021) was proposed, following a similar design principle as Point Transformer. However, PCT employs global attention rather than local attention, which restricts its scalability when applied to large-scale scenes with a high volume of points.

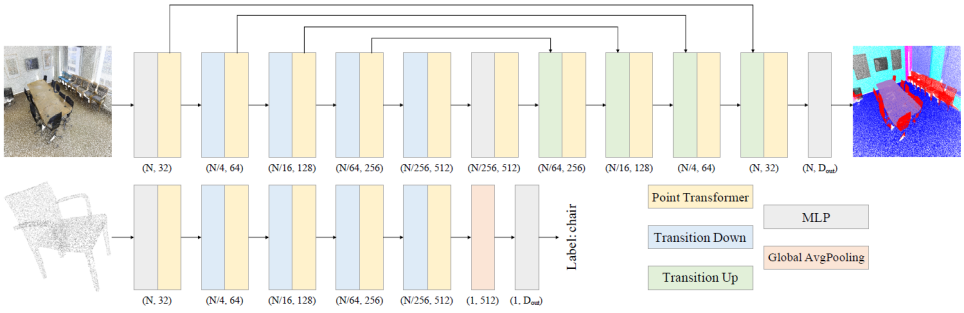


Figure 2.6: Point Transformer architecture (H. Zhao *et al.*, 2021).

Following Point Transformer, many research efforts have been devoted to 3D transformer architectures for point cloud understanding and scene segmentation. PT v2 (X. Wu *et al.*, 2022) improves Point Transformer by using grouped weighting techniques and partition-based feature pooling. Stratified Transformer (Lai *et al.*, 2022) enhances the capture of contextual dependencies in long ranges by simultaneously sampling dense nearby points and sparse distant points as keys. Superpoint Transformer (Robert *et al.*, 2023) hierarchically partitions the input point set into superpoints, which are groups of points that are spatially proximate and geometrically homogeneous. A transformer model is then applied to capture the relationships between these partitioned superpoints at multiple scales. Similarly, P.-S. Wang (2023) revisited the Octree data structure, proposing an Octree-based transformer to enhance the efficiency of point cloud interpretation. Y.-Q. Yang *et al.* (2023) extended the successful image-based Swin Transformer to 3D, showing that a pretrained Swin3D model on large synthetic datasets can generalize well to downstream segmentation tasks. PT v3 (X. Wu *et al.*, 2024) significantly

advances the previous Point Transformer and PT v2 by replacing the traditional kNN search with an efficient serialized neighbor mapping of points organized with specific spatial patterns, thereby enabling larger receptive fields.

2.2.4. SUMMARY

To conclude, 3D semantic segmentation of LiDAR point clouds is a valuable yet complex task that requires a comprehensive understanding of point data structure, object geometry, and spectral characteristics. Recent state-of-the-art deep learning methods have largely advanced this field, negating the need to manually design and select handcrafted point features. However, unlike 2D image processing, deep learning for point clouds is still rapidly evolving. In this research, we address 3D semantic segmentation through two primary approaches: Locally, we incorporate boundary priors to encourage consistent semantic segmentation near object boundaries (Chapter 3); Globally, we perform class prototype expansion to enhance the overall representation of point features (Chapter 4). Through these two techniques, our objective is to achieve more accurate and consistent semantic segmentation of urban point clouds.

2.3. 3D INSTANCE SEGMENTATION

Compared to 3D semantic segmentation, 3D instance segmentation is a more challenging task, as it requires a finer-grained understanding of point clouds. Specifically, it not only needs to classify points into different semantic categories but also needs to distinguish individual object instances within the same semantic category. Most research in this domain leverages the networks discussed in Section 2.2 for deep point feature embedding, then adopts additional network modules or postprocessing techniques to achieve instance-level segmentation. Overall, existing studies in 3D instance segmentation can be divided into proposal-based methods and proposal-free methods (Y. Guo *et al.*, 2020).

2.3.1. PROPOSAL-BASED METHODS

Proposal-based methods tackle instance segmentation in two stages: object detection and instance masking. This *detect-then-segment* strategy was initially introduced in Mask Region-based Convolutional Neural Networks (Mask R-CNN) (He *et al.*, 2017) and later became highly influential in image-based instance segmentation tasks (Cai & Vasconcelos, 2018; Z. Huang *et al.*, 2019; X. Wang *et al.*, 2020).

Inspired by the success of Mask R-CNN and its variants in the 2D domain, proposal-based approaches in 3D also adopt a top-down strategy, wherein proposals are first generated and then refined to segment object instances. Yi *et al.* (2019) introduced the Generative Shape Proposal Network, which employs an analysis-by-synthesis strategy to obtain high-quality 3D proposals. A region-based

PointNet is then utilized for proposal refinement and instance mask prediction. Hou *et al.* (2019) presented the 3D-SIS network to jointly learn from both 2D and 3D modalities, combining multi-modal features within deep neural layers to achieve accurate 3D instance segmentation. Different from them, 3D-BoNet (B. Yang *et al.*, 2019) directly predicts object bounding boxes by regressing box boundaries using MLP layers (Figure 2.7), followed by a point masking branch to obtain instance masks. F. Zhang *et al.* (2020) introduced a specialized instance segmentation network designed for large-scale outdoor LiDAR point clouds, which learns a dense feature representation on the bird's eye view, allowing more accurate localization and segmentation, particularly for small and far-away objects.

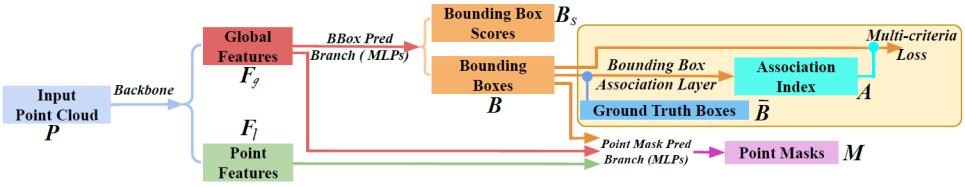


Figure 2.7: 3D-BoNet workflow (B. Yang *et al.*, 2019).

Overall, proposal-based methods share a simple and straightforward idea. Adopting the *detect-then-segment* strategy, they can generate instance proposals with high objectiveness. Nevertheless, most methods require multi-stage training and extensive postprocessing to achieve good instance segmentation results.

2.3.2. PROPOSAL-FREE METHODS

Contrary to proposal-based approaches, most proposal-free methods rely on a *predict-then-group* pipeline. The main assumption is that points belonging to the same object instance exhibit highly similar features, and thus can be grouped based on discriminative feature learning.

Similarity Group Proposal Network (W. Wang *et al.*, 2018) is one of the pioneering works in this category, learning a similarity map between pairwise points in the feature space to group object instances. However, this method is limited by its high computational and memory resource requirements. X. Wang *et al.* (2019) introduced the Associatively Segmenting Instances and Semantics network to simultaneously learn semantic and instance features, which mutually enhance each other. To encourage individual points to closely group into the same instance, PointGroup (L. Jiang *et al.*, 2020) proposed an offset prediction module that learns explicit offset vectors for each point directing towards its instance centroid. This technique enables robust point grouping and has been widely adopted in follow-up studies. S. Chen *et al.* (2021) extended PointGroup by adding a hierarchical instance aggregation scheme. Dong *et al.* (2022) further enhanced consistency in instance segmentation through the use of regional purity measurements. In contrast, SoftGroup (Vu *et al.*,

2022) performs soft grouping by assigning each point to multiple instances and refines the instance segmentation using a top-down strategy.

Figure 2.8 illustrates the pipeline of the representative proposal-free method, PointGroup. Unlike proposal-based methods, these methods do not require additional object detection modules. However, their point-wise grouping strategies often lead to imprecise object boundaries and low objectness. A number of works also seek transformer-based architectures to address 3D instance segmentation. Mask3D (Schult *et al.*, 2023) represents individual object instances as queries and employs transformer decoding layers to generate instance masks. Lu *et al.* (2023) further introduced a query refinement module that integrates both superpoint features and multiscale features, enabling high-coverage, low-redundancy object query initialization. Oneformer3D (Kolodiazhnyi *et al.*, 2024) investigates the implicit relationship between semantic segmentation and instance segmentation, unifying the two tasks within a single framework. Despite their promising performance, transformer-based approaches remain computationally and memory-intensive, posing challenges for large-scale point cloud processing.

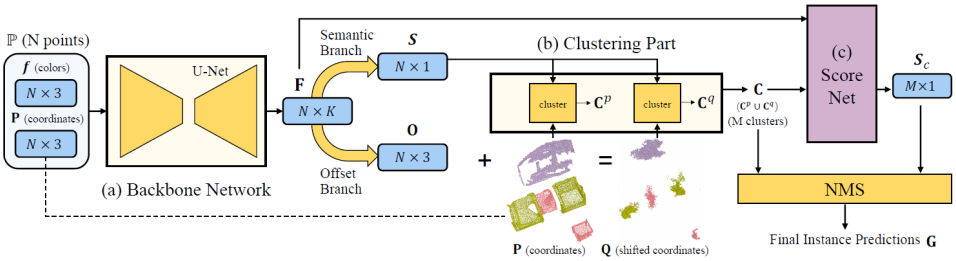


Figure 2.8: PointGroup pipeline (L. Jiang *et al.*, 2020).

2.3.3. OBJECT-SPECIFIC INSTANCE SEGMENTATION

The methods discussed in previous Sections 2.3.1 and 2.3.2 were mainly proposed for general 3D instance segmentation tasks. Besides, depending on specific research objectives, instance segmentation techniques can also be tailored for particular urban object categories. This section explores two distinct object types, buildings and trees, which constitute the majority of 3D urban environments and are important objects in urban applications. Precise mapping and analysis of building and tree instances facilitate urban digitalization, simulations, enhanced environmental monitoring, infrastructure management, and ecological assessments, supporting a wide range of urban development initiatives. Nevertheless, compared to buildings, trees are more challenging urban objects due to their irregular structures and complex geometries.

Buildings are man-made objects that typically exhibit regular shapes, planar surfaces, and sharp edges. These characteristics enable automated extraction of building outlines, also referred to as footprints, which play a crucial role in

segregating individual 3D building instances. Widyaningrum *et al.* (2020) utilized medial axis transform descriptors to detect building corner points, which were subsequently connected to delineate building outlines. D. Chen *et al.* (2017) detected building boundary primitives by clustering and assembled these primitives by taking into account topological consistencies. Rottmann *et al.* (2022) formulated the building footprint extraction problem as a Traveling Salesperson Problem, optimizing the shortest possible tour among points to derive individual building boundaries. Besides, more recent studies also seek up-to-date machine learning and deep learning methods for building footprint extraction from point clouds (Dabove *et al.*, 2024; Park & Guldmann, 2019; M. Sharma & Garg, 2023). Given the availability of building footprint data, one can easily extrude and model individual building objects. Studies of 3DBAG (Peters *et al.*, 2023) and City3D (J. Huang *et al.*, 2022) have reconstructed massive high-quality 3D building instances utilizing footprint information,

In contrast to buildings, trees present greater challenges for instance segmentation due to their irregular shapes and the structural complexity of forest ecosystems. An analogy approach to building footprint extraction is *tree watershed segmentation*, which delineates tree crown contours based on height variations in point cloud data (Beucher, 1979; Q. Chen *et al.*, 2006). This method typically requires transforming the raw data into Digital Surface Models (DSMs) or Canopy Height Models (CHMs) to facilitate crown delineation. An alternative approach, *tree point clustering*, operates directly on raw point clouds by identifying dense clusters in 3D space and segmenting individual tree instances (Hakula *et al.*, 2023; Malladi *et al.*, 2024; J. Wang *et al.*, 2018). Recently, deep learning techniques have frequently been used in 3D tree instance segmentation, either on transformed data such as DSMs and CHMs (Chang *et al.*, 2022; J. Wang *et al.*, 2019), or on the raw vegetation point clouds (Henrich *et al.*, 2024; T. Jiang, Wang, *et al.*, 2023; H. Luo *et al.*, 2021). They have demonstrated great potential for large-scale forest monitoring and ecological studies.

2.3.4. SUMMARY

Compared to 3D semantic segmentation, 3D instance segmentation provides a more granular understanding of urban scenes by distinguishing distinct object instances within the same semantic category. Many instance segmentation methods build upon foundational point cloud learning networks, as discussed in Section 2.2, to extract robust point features. Then, they adopt additional modules such as grouping and masking to obtain final object instances.

Besides, a large number of 3D instance segmentation approaches have also been tailored for specific urban object types, particularly buildings and trees, as these elements constitute the majority of urban environments. Buildings typically exhibit regular geometric structures and sharp edges, which facilitate the automated extraction of building footprints and the delineation of individual instances. In contrast, trees are significantly more challenging to segment due to their irregular

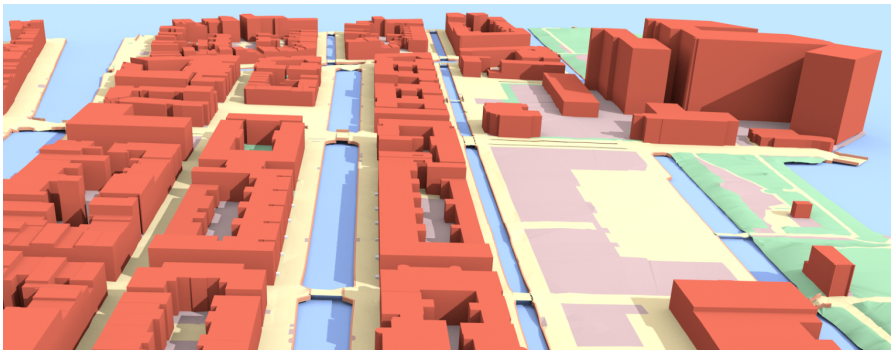
geometries, varying types, and the structural complexity of forestry systems. These factors often lead to segmentation errors, such as oversegmentation or undersegmentation, in existing approaches. In this thesis, we focus on 3D instance segmentation of urban trees, motivated by their critical ecological and environmental importance. Our goal is to develop a robust and scalable method for large-scale 3D tree instance segmentation, capable of handling occlusions, overlaps, and varying tree morphologies (Chapter 5).

2.4. APPLICATIONS OF SEGMENTED POINT CLOUDS

Recent developments in remote sensing and laser scanning technologies have significantly encouraged the fast, reliable, and low-cost acquisition of 3D LiDAR point clouds. In particular, the proliferation of publicly available LiDAR datasets has greatly accelerated deep learning research in point cloud processing, interpretation, and segmentation, which serves as a foundation for various downstream applications in urbanism and computer vision. These advancements enable precise and automated urban environment assessments, contributing to digital innovations in urban planning, governance, and socio-ecological studies.

Semantically enriched point clouds can be used to generate base maps in cartography and cadastral applications, as they provide detailed representations of key urban features such as buildings, vegetation, roads, and bridges. When integrated with satellite imagery or aerial photos, they enable a comprehensive and up-to-date analysis of land cover, land use, infrastructure, and green space distribution (Kadaster, 2025). Specifically, unlike 2D images, segmented point clouds offer precise 3D measurements of object surfaces, thereby facilitating the accurate modeling and reconstruction of urban environments. A large number of studies have successfully reconstructed high-fidelity 3D building (J. Huang *et al.*, 2022; Ledoux *et al.*, 2021; Nan & Wonka, 2017; Peters *et al.*, 2023) and tree models (Du *et al.*, 2019; Fan *et al.*, 2020; Y. Liu *et al.*, 2021) from segmented urban point clouds. Figure 2.9 presents two representative approaches for urban building and tree reconstruction.

By incorporating external data sources and domain expertise from other fields, more advanced urban applications can be performed. For example, well-segmented and reconstructed buildings support quantitative analyses of visibility (L. Liu *et al.*, 2010), daylight (Nazari & Matusiak, 2024), and solar potential (D. Li *et al.*, 2015), contributing to energy-efficient building designs and sustainable urban planning (Agugiaro, 2016). These reconstructions also facilitate cost-effective Building Information Modeling (BIM) (Arroyo Ohori *et al.*, 2017). Similarly, precise mapping and inventory of individual trees are essential for assessing above-ground biomass, forest productivity, and carbon storage (Fan *et al.*, 2020; Le Toan *et al.*, 2011; F. Zhang *et al.*, 2016), which aid policymakers in developing climate change mitigation strategies. The integration of key urban features (e.g., buildings, trees, roads, and civil structures) further enables the automated digitalization of urban environments, facilitating applications such as navigation (Cappelle *et al.*, 2012), autonomous



(a) 3D reconstructed buildings in Leiden.

(b) 3D tree modeling. From left to right: input points, tree branches, final model.

Figure 2.9: Reconstructing 3D urban buildings (Ledoux *et al.*, 2021) and trees (Du *et al.*, 2019) from point clouds.

driving (Yurtsever *et al.*, 2020), and traffic detection (Fu *et al.*, 2023). Additionally, by incorporating multidisciplinary expertise, researchers can perform advanced urban simulations and environmental monitoring, including Computational Fluid Dynamics analysis (Philips, 2014), air pollution modeling (Ujang *et al.*, 2013), wind flow simulation (Pađen *et al.*, 2022), flood risk assessment (P. Luo *et al.*, 2022), and disaster management (Kemeç & Duzgun, 2006).

In conclusion, 3D point cloud data with enriched semantic and instance-level information greatly benefits a broad array of applications in computer vision, urban planning and design, as well as ecological studies, enabling fast, comprehensive, and in-depth analysis of the urban environment, which further assists decision-making in fields such as smart city development, urban policy formulation, and natural resource management. Additionally, integrating these data with external information and domain-specific expertise supports advanced socio-ecological analyses, allowing for more precise modeling of complex urban systems. These capabilities ultimately contribute to the development of sustainable cities and the improvement of urban living conditions.

3

LOCAL BOUNDARY-GUIDED SEMANTIC SEGMENTATION

This chapter addresses 3D semantic segmentation from a local perspective. While deep learning has achieved remarkable success in segmenting and interpreting urban scenes from point clouds, most learning-based methods, particularly point CNN methods, still severely suffer from inconsistencies in segmentation near local object boundaries. To cope with this limitation, this chapter proposes a novel boundary-guided feature propagation approach that employs a multitask learning strategy to explicitly guide the boundaries to their original locations. With one shared encoder, our network simultaneously performs boundary localization, directional prediction towards object interiors, and semantic segmentation. Then, we integrate the predicted boundaries and directions to propagate the learned features, enhancing segmentation accuracy. Unlike previous studies that rely on post-processing or implicit feature encoding, our method offers an explicit, end-to-end, and single-stage solution. We have conducted extensive experiments on the indoor S3DIS and outdoor SensatUrban datasets against various baseline methods, demonstrating that the proposed method yields consistent improvements by reducing boundary errors.

This chapter is based on the paper: Shenglan Du, Nail Ibrahimli, Jantien Stoter, Julian F.P. Kooij, Liangliang Nan. Push-the-boundary: Boundary-aware feature propagation for semantic segmentation of 3d point clouds. 2022 IEEE international conference on 3D vision (3DV). IEEE, 2022. DOI: 10.1109/3dv57658.2022.00025.

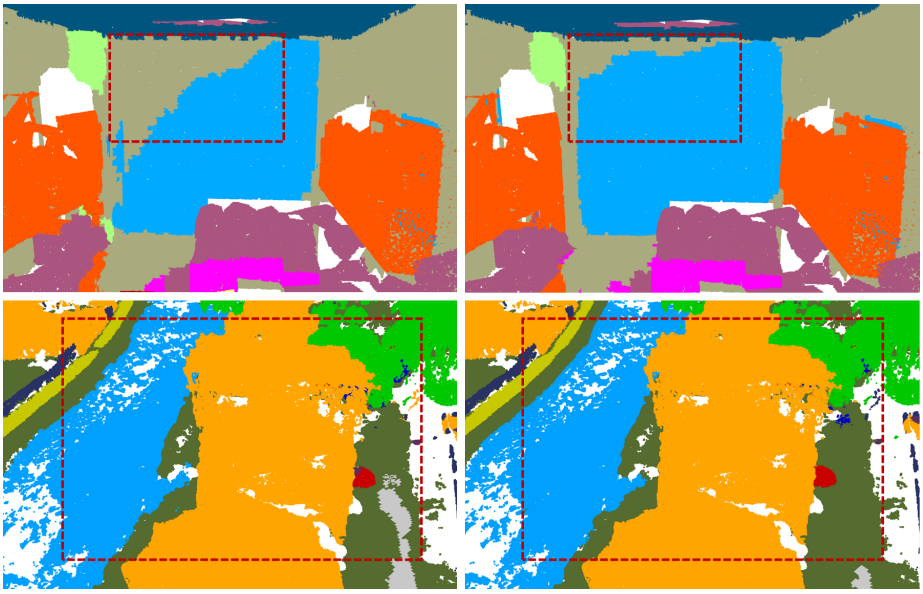
3.1. INTRODUCTION

3D semantic segmentation of point clouds is a fundamental yet challenging task that aims to assign a semantic category (e.g., building, tree, window) to each point in the scene. Accurate semantic interpretation of 3D point clouds is essential for numerous applications across urban planning, ecological monitoring, environmental analysis, computer vision, and robotics.

Driven by the widespread success of deep learning networks, specifically CNNs, in 2D image recognition, extensive research has been conducted on 3D semantic segmentation of point clouds. As introduced in Section 2.2, early studies focused on transforming points into regular grids or voxels as input to standard CNNs (H. Guo *et al.*, 2016; Maturana & Scherer, 2015; Riegler *et al.*, 2017; H. Su *et al.*, 2015; Z. Wu *et al.*, 2015; Z. Yang & Wang, 2019). However, these methods introduce extra computational costs and information loss, which often lead to suboptimal segmentation performance. To avoid that, the seminal PointNet (Qi, Su, *et al.*, 2017) directly consumes point clouds and extracts point features through a sequence of shared MLP layers. Following PointNet, many point-based deep learning frameworks have been introduced (Q. Hu *et al.*, 2020; Y. Li *et al.*, 2018; Thomas *et al.*, 2019; S. Wang *et al.*, 2018).

Despite their strong performances in 3D semantic segmentation, most point-wise feature learning networks suffer from an often-overlooked limitation in feature propagation: the loss of local information in decoding. Specifically, since most networks adopt CNN-like architectures with the encoding-decoding strategy, the existence of pooling layers in the encoder can capture hierarchical semantic features with increased receptive fields. While this facilitates object-level recognition, it also produces coarse feature representations at lower resolutions. During decoding, these coarse features are then propagated back to the original resolution using nearest-neighbor upsampling, which ignores point-level variations among different semantic categories. As a result, networks lose object boundary details and fail to generate accurate predictions.

A number of studies have made fruitful attempts to refine semantic segmentation at local levels. One attempt is to model contextual affinity using graphical models such as Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) (L.-C. Chen *et al.*, 2017; Zheng *et al.*, 2015). However, MRFs and CRFs are inserted as additional modules, which are difficult to integrate into end-to-end network training. Considering that boundaries naturally indicate the transition between objects of different semantic categories, another line of research leverages boundary information to enhance segmentation (Ding *et al.*, 2019; Gong *et al.*, 2021; Hayder *et al.*, 2017; Z. Hu *et al.*, 2020; Zhen *et al.*, 2020). Most of these methods introduce boundary detection as an auxiliary task for semantic segmentation. With one shared encoder, the two tasks implicitly improve each other. However, despite their benefits, these approaches do not explicitly address semantic segmentation. Meanwhile, they require extra encoding layers to fuse features from the two tasks, which is more challenging for the network to optimize as more parameters are involved.



(a) Segmentation results of baseline (b) Segmentation results of ours

Figure 3.1: Comparison between the segmentation results from the baseline network (Thomas *et al.*, 2019) and ours on both indoor (top) and outdoor (bottom) scenes.

Contrary to existing works, our motivation is to refine semantic segmentation by explicitly pushing the boundary towards desired directions. Predicting a direction scheme to refine semantic segmentation has been recently studied in 2D image analysis (Mazzini & Schettini, 2019; Yuan *et al.*, 2020). Nevertheless, in these studies, directions solely serve as post-processing tools to adjust predicted semantic labels. To better exploit boundary and directional priors for network feature learning, we propose a novel end-to-end framework that integrates boundary detection, direction prediction, and semantic segmentation. As illustrated in Figure 3.1, standard point-based CNNs tend to lose boundary details at the object level, whereas our approach effectively preserves local details and refines segmentation along object boundaries. The proposed network consists of a single feature encoder and jointly produces three streams of point-wise predictions: (i) a boundary label that gives a binary prediction indicating whether a point belongs to an object boundary, (ii) a direction vector originating from the nearest boundary and pointing toward the object's interior, and (iii) a semantic class label. We demonstrate that even though the network is primarily optimized for semantic segmentation, it inherently learns discriminative features that also benefit boundary detection and direction prediction, leading to more accurate and consistent segmentation performance.

The key to our approach is a lightweight guiding mechanism that effectively fuses the boundary and direction priors to refine the segmentation. The main insight

behind this strategy is to mitigate the loss of local information in network decoding by leveraging predicted directional cues (i.e., pointing from boundary to interior) to guide feature propagation. By doing so, our method prevents feature mixing between different semantic categories and ensures that segmentation boundaries are explicitly refined along the desired directions.

Our contributions can be summarized in two-fold:

- We propose a novel end-to-end network for joint semantic segmentation, boundary detection, and direction prediction to enhance the interpretation of 3D urban point clouds. The tasks of boundary detection and direction prediction can appropriately improve segmentation output.
- We introduce a new boundary-aware feature upsampling strategy that directs feature propagation along predicted boundary-aware directions. This mechanism can be seamlessly incorporated into existing segmentation frameworks, providing a generalizable enhancement for 3D semantic segmentation tasks.

3.2. RELATED WORK

As discussed in Chapter 2, the rapid emergence and development of deep learning have drastically revolutionized 3D semantic segmentation of point clouds. Specifically, a large volume of point-based deep learning networks has been introduced to learn high-level semantics directly from raw point cloud data (Section 2.2). Nevertheless, many of these networks exhibit segmentation noises and inconsistencies near local object boundaries. Therefore, in this section, we review the methods specifically designed to enhance local segmentation accuracy and boundary refinement.

3.2.1. GRAPH-BASED SEGMENTATION

Graph-based approaches utilize the inherent structure of graphs to represent and analyze contextual relationships among individual data points, effectively modeling local geometric properties. By embedding node features and propagating contextual information through the graph's topological connections, these approaches iteratively refine object predictions and encourage locally consistent segmentation outputs (D. Xu *et al.*, 2017).

Simonovsky and Komodakis (2017) introduced Edge-Conditioned Convolution that generalizes the convolution operator from regular grids to arbitrary graphs, allowing it to handle graphs of varying size and connectivity. Following that, Landrieu and Simonovsky (2018) proposed SuperPoint Graph (SPG), which constructs a graph on over-segmented point sets to capture contextual relationships between object components. L. Jiang *et al.* (2019) designed a point-edge interaction network that hierarchically integrates point features into edge features to achieve locally consistent segmentation. Besides, graphical models such as MRFs and CRFs have

been incorporated into deep networks to refine segmentation results (L.-C. Chen *et al.*, 2017; Zheng *et al.*, 2015). Based on the fact that spatially adjacent pixels or points presenting similar features are likely to share the same semantic labels, MRF- and CRF-based methods formulate segmentation as a probabilistic inference problem, leading to smoother and more consistent segmentation, particularly near object boundaries. However, integrating additional MRF and CRF modules adds extra computational costs to the network optimization process.

3.2.2. BOUNDARY-AWARE REFINEMENT

Standard CNN-based networks for semantic segmentation are limited in their ability to model point-level accurate object boundaries, mainly due to the significant loss of local-level features in the feature encoding and decoding layers. Besides using graphical models to softly encourage local segmentation consistency, many research studies have been proposed to directly refine and sharpen the object boundaries.

In 2D image processing, Boundary Neural Field (Bertasius *et al.*, 2016) formulates a global energy model to enhance semantic segment coherence with predicted boundary cues. Other works combine the semantic segmentation task and the boundary detection task into a unified network (Ding *et al.*, 2019; Hayder *et al.*, 2017; Y. Liu *et al.*, 2017; J. Su *et al.*, 2019; Takikawa *et al.*, 2019; Z. Yu *et al.*, 2017). In these networks, the two tasks share a common feature encoder and are expected to mutually improve each other. In the 3D domain, boundary-aware strategies have also been explored. JSENet (Zhen *et al.*, 2020) incorporates boundary detection into 3D point cloud semantic segmentation. However, this approach relies on additional feature enhancement modules and requires a curriculum learning strategy for optimal segmentation performance. Several studies (Gong *et al.*, 2021; M. Xu, Zhou, *et al.*, 2021) address 3D boundary detection through adaptive methods or preprocess modules. To better delineate object boundaries, Tang *et al.* (2022) introduced a local contrastive loss to contrast the features across the scene boundaries.

Unlike previous studies, our work focuses on the propagation of interior information during network decoding. We show that the guided feature propagation naturally recovers information near boundaries, without the need for external constraints or additional training modules. Meanwhile, our method does not treat boundaries as intermediate outputs. It jointly learns boundary cues for guiding the feature propagation, which can be seamlessly integrated into existing networks.

3.3. METHOD

Our network consists of one feature encoder followed by three task streams:

- **Boundary detection**. For each point, we predict a binary label to indicate whether the point lies on the object boundary;
- **Direction prediction**. For each point, we predict a direction vector from its

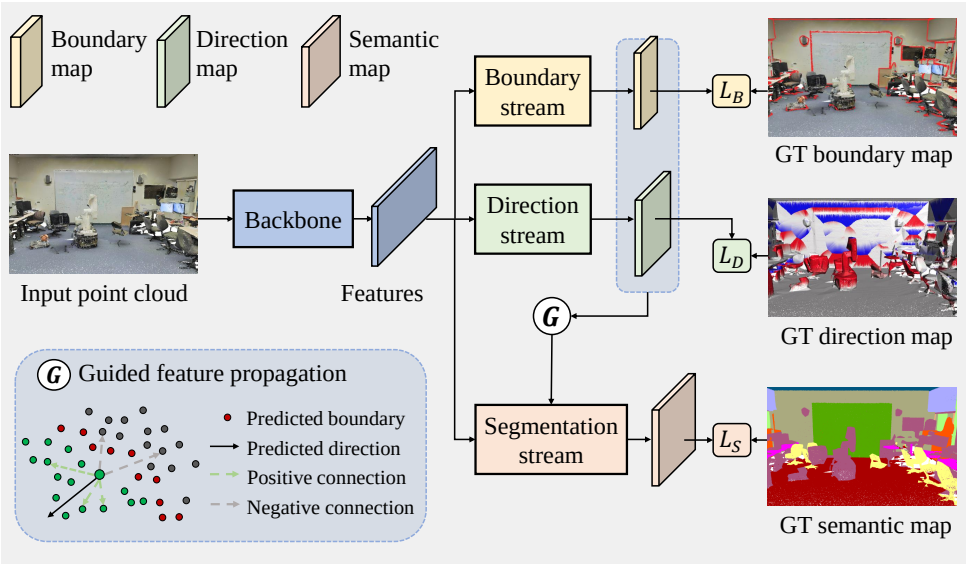


Figure 3.2: Network architecture overview. Our method can adopt various feature encoders, such as KP-Conv (Thomas *et al.*, 2019) and PointNet++ (Qi, Yi, *et al.*, 2017). GT is the Ground Truth. L_B, L_D, and L_S are the network losses for the boundary, direction, and segmentation streams, respectively. In the guided feature propagation, a positive connection means the vector of the point pair has a positive cosine angle with the predicted direction, while a negative connection means a negative cosine angle.

closest boundary to the interior of the object;

- **Semantic segmentation**. We give a semantic category prediction per point.

The boundary and directional predictions are then integrated to guide feature propagation in a boundary-aware manner. Figure 3.2 gives an overview of our framework. In the following sections, we provide a detailed explanation of each part.

3.3.1. NETWORK ARCHITECTURE

We primarily use KP-Conv (Thomas *et al.*, 2019) as our backbone network. It represents the state-of-the-art in point-based CNN networks, extending the convolutional operator from 2D to 3D by defining convolutional kernels as evenly distributed 3D points within a local spherical space. However, due to its encoding-decoding strategy that leads to information loss at finer levels, KP-Conv experiences significant errors near local boundaries (Figure 3.1a). To further verify the capacity to extend our approach to other point learning networks, we also conduct segmentation experiments using PointNet++ (Qi, Yi, *et al.*, 2017) in Section 3.4.4.

BOUNDARY DETECTION STREAM

Given an input point cloud $P \in R^{N \times F}$, where N is the number of points and F is the input feature dimension, we use the boundary detection stream to predict a binary map $B \in R^{N \times 2}$. In this map, a value of 1 indicates that a point lies on a boundary, while 0 corresponds to points located within the object's interior. It is important to note that we focus on semantic boundaries, which refer to the boundaries between different semantic categories. Since boundary points only account for a small portion of the entire point set, we employ a weighted binary cross-entropy loss to supervise this task.

DIRECTION PREDICTION STREAM

The use of direction prediction schemes to refine semantic segmentation has been recently studied (Mazzini & Schettini, 2019; Yuan *et al.*, 2020) in 2D image processing. Specifically, Yuan *et al.* (2020) jointly localizes the boundary pixel and predicts the direction from boundary pixels to their corresponding interior pixels. Motivated by the empirical observation that label predictions of interior pixels are generally more reliable, this method replaces the initially unreliable boundary pixel predictions with those of the associated interior pixels. However, it requires separate learning for direction vectors, which are then used as a post-processing step to refine segmentation outputs.

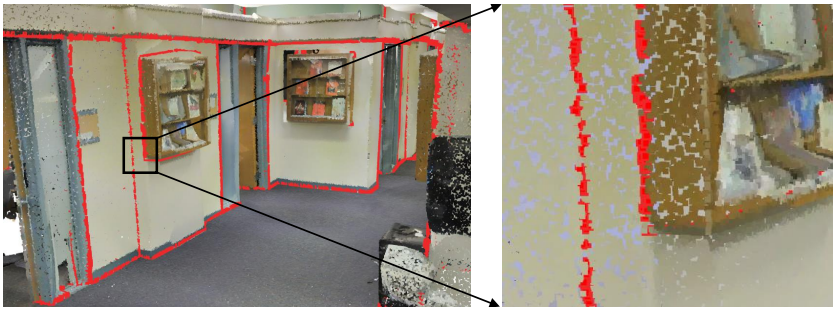
In this study, we extend the direction prediction scheme to 3D semantic segmentation and integrate it into an end-to-end learning framework. We use the direction stream to predict a direction map $D \in R^{N \times 3}$, with each row (d_x, d_y, d_z) representing a unit direction vector pointing from the nearest boundary toward the object's interior. By designing D in this manner, each point is assigned a direction that guides it toward the homogeneous interior region. The predicted directions serve as an effective indicator for network message passing, which, when combined with boundary predictions, facilitates feature propagation in the semantic segmentation stream.

SEMANTIC SEGMENTATION STREAM

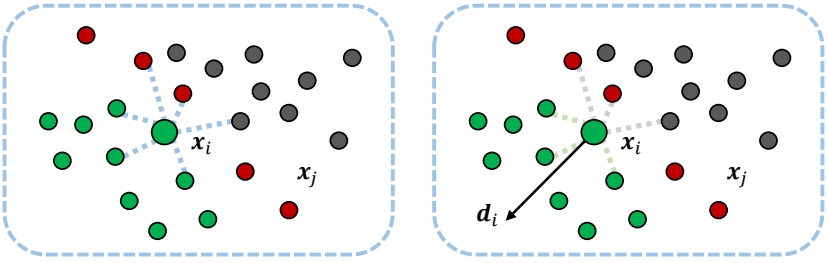
For the point cloud $P \in R^{N \times F}$, the semantic segmentation stream outputs a semantic mask $S \in R^{N \times K}$, where K is the total number of semantic categories. This stream predicts a probability distribution over the K categories for each point, enabling per-point semantic classification.

3.3.2. BOUNDARY-GUIDED FEATURE PROPAGATION

Point-wise learning networks, particularly point CNN networks, employ pooling layers to progressively sub-sample points, capturing high-level semantic features in the latent space. During the subsequent decoding stage, features are propagated from



(a) Drastic semantic transitions happen across the object boundaries.



Standard feature propagation Guided feature propagation

(b) Two ways of feature propagation.

Figure 3.3: Feature propagation near object boundaries. We propagate features from neighboring points $\mathbf{x_j}$ in the higher layer to the central point $\mathbf{x_i}$ in the lower layer. Standard feature propagation ignores object boundaries, represented as red dots, resulting in the mixed feature representation of $\mathbf{x_i}$. In contrast, our proposed strategy uses the predicted direction vector $\mathbf{d_i}$ to guide feature propagation. Features from neighboring $\mathbf{x_j}$ aligned with $\mathbf{d_i}$ are encouraged to propagate to the central $\mathbf{x_i}$ (green dashed lines), while feature propagation from the opposing neighbors is constrained (grey dashed lines).

sub-sampled points back to the original points. A common propagation strategy is inverse distance interpolation within a local kNN region. For example, PointNet++ Qi, Yi, *et al.*, 2017 uses $k = 3$ for feature propagation, while KP-Conv Thomas *et al.*, 2019 uses $k = 1$, thus reducing to the nearest neighbor feature upsampling. Standard feature propagation assumes that spatially adjacent points exhibit semantic similarity. However, it ignores drastic semantic transitions across object boundaries. Figure 3.3a visualizes an example where a semantic shift occurs from a wall to a bookshelf at boundary points. Due to such semantic transitions, points at the boundary may aggregate features from different objects, leading to ambiguous feature representations that can degrade the accuracy of the final segmentation.

As shown in Figure 3.3b, to mitigate the issue of blurred feature representations near object boundaries, we incorporate predicted boundary information and

directional cues to guide feature propagation within the network's decoding layers. The core of our approach lies in encouraging features to propagate along the desired direction for generating more purified and spatially coherent feature maps.

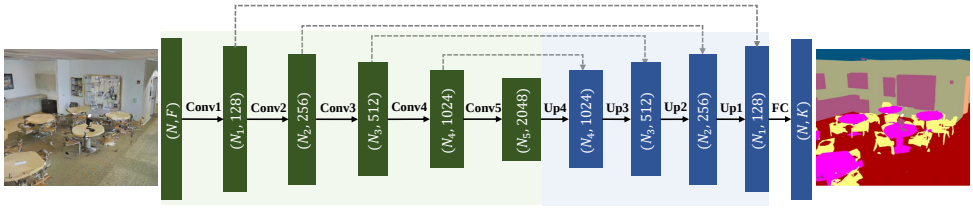


Figure 3.4: Network architecture details, adopting KP-Conv as the backbone. Only the segmentation stream is visualized for clarity. *Conv* denotes a convolutional-like operation for feature aggregation. *Up* denotes feature upsampling. *FC* is a fully connected layer. The grey dashed lines denote the skipped links. N is the number of points, F the dimension of the input feature, K the number of output categories, and N_i the number of points in each layer, with the initial layer satisfying $N_1 = N$.

During network decoding, features are upsampled from sparser points in deeper layers to denser points in shallower layers (Figure 3.4). Specifically, features from points $\mathbf{x_j}$ in l^{th} layer are propagated to points $\mathbf{x_i}$ in $l-1^{th}$ layer. This feature propagation is conducted under

$$f^{l-1}(\mathbf{x_i}) = \sigma(\Phi(\frac{\sum_{j=1}^{k} w(\mathbf{x_j}) f^{l}(\mathbf{x_j})}{\sum_{j=1}^{k} w(\mathbf{x_j})}) \oplus f_{skip}^{l-1}(\mathbf{x_i})), \tag{3.1}$$

where $\mathbf{x} \in R^3$ denotes a 3D point and k is the total number of neighboring points. The decoder features $f^{l}(\mathbf{x_j})$ at l^{th} layer corresponding to neighboring points $\mathbf{x_j}$ are first interpolated based on associated weights w, and then concatenated with skip-linked encoder features, which are then processed through an MLP layer and a Rectified Linear Unit (ReLU) activation function to obtain the new features $f^{l-1}(\mathbf{x_i})$ for points $\mathbf{x_i}$. Here, f_{skip} is the skip-linked features, $\Phi(\cdot)$ denotes an MLP operator, and $\sigma(\cdot)$ is the ReLU activation function.

Taking into account the predicted boundaries and directions, we compute an adaptive weight term w to guide feature propagation, i.e.,

$$\begin{aligned} w(\mathbf{x_j}) &= \max(0, w_s(\mathbf{x_j}, \mathbf{x_i}) + \alpha\, w_c(\mathbf{x_j}, \mathbf{x_i})), \\ w_s(\mathbf{x_j}, \mathbf{x_i}) &= \exp\frac{-||\mathbf{x_j} - \mathbf{x_i}||_2}{r}, \\ w_c(\mathbf{x_j}, \mathbf{x_i}) &= \exp(P_b(\mathbf{x_i}) - 1)\cos(\mathbf{x_j} - \mathbf{x_i}, \mathbf{d_i}), \end{aligned} \tag{3.2}$$

where we linearly combine the spatial similarity w_s and cosine similarity w_c between $\mathbf{x_j}$ and $\mathbf{x_i}$. The spatial similarity weight w_s is determined by the Gaussian distance between the given pair of points, whereas the cosine similarity weight w_c quantifies the angular deviation between the spatial displacement vector and the predicted directions. $P_b(\mathbf{x_i})$ represents the predicted boundary probability of point $\mathbf{x_i}$, $\mathbf{d_i}$ is its predicted direction, and $\cos(\cdot)$ computes the cosine similarity between two vectors. $\max(\cdot)$ ensures non-negative weights by eliminating negative values. The constant coefficients r and α are used to balance the two similarity terms.

To maintain the consistency of feature propagation, the computed weights are further normalized using L_1 normalization, i.e.,

$$w(\mathbf{x_j}) = \frac{w(\mathbf{x_j})}{\sum_{j=1}^{k} w(\mathbf{x_j})}. \tag{3.3}$$

Our propagation strategy brings several benefits to semantic segmentation. First, for points near object boundaries predicted with high predicted boundary probabilities (e.g., $P_b \approx 1$), we assign greater importance to the neighboring points that align with the predicted direction, while we suppress the influence of oppositely oriented neighbors. This mechanism naturally encourages features to propagate along the desired direction, enhancing boundary preservation. Conversely, for interior points with low boundary probability predictions (e.g., $P_b \approx 0$), the cosine similarity term w_c is significantly reduced. Feature propagation is primarily governed by spatial similarity, where all neighboring points contribute based on their distances to the centroid. As a result, we facilitate consistent segmentation within object interiors. Furthermore, the proposed weight term is formulated as a continuous function of the boundary prediction map B and the direction prediction map D, which guarantees smooth gradient backpropagation of the network.

In Section 3.4.6, we present ablation studies to validate the effectiveness of the proposed boundary-guided feature propagation mechanism.

3.3.3. NETWORK SUPERVISION

Our network is designed to perform three tasks, as introduced in Section 3.3.1, with each task stream supervised by a distinct loss term. In this section, we provide a detailed formulation of the loss terms corresponding to their respective tasks.

Boundary detection loss. Boundary detection is a binary classification task. Given the sparsity of boundary points in the dataset, we employ the weighted binary cross-entropy loss to mitigate class imbalance using

$$L_B = -\sum_{i=1}^{N} [\beta \hat{b_i} \log b_i + (1-\beta)(1-\hat{b_i}) \log(1-b_i)], \tag{3.4}$$

where $\hat{b_i}$ represents the Ground Truth (GT) binary label, with $\hat{b_i} = 1$ for boundary

points and $\hat{b_i} = 0$ for interior points. b_i is the network softmax output of the i^{th} point. We use a coefficient β to balance the boundary class and the object's interior class.

Direction prediction loss. In the previous study of 2D image segmentation (Yuan *et al.*, 2020), discrete directions are predicted by partitioning the full angular space into a predefined set of directional bins. Thus, it restricts the prediction to a limited set of directions. Unlike that, we formulate direction prediction as a regression task, enabling the model to predict continuous directional embeddings across the entire 3D space. Our predictions are more adaptable for complex 3D scenes. To supervise this stream, we adopt Mean Squared Error (MSE) loss, which is defined as follows:

$$L_D = \sum_{i=1}^{N} \|\mathbf{d_i} - \hat{\mathbf{d}}_\mathbf{i}\|_2^2, \tag{3.5}$$

where $\mathbf{d_i} \in R^3$ is the predicted direction, and $\hat{\mathbf{d}}_\mathbf{i} \in R^3$ is the GT direction. Both directions have a unit magnitude. We also evaluated an alternative loss formulation based on the dot product of $\mathbf{d_i}$ and $\hat{\mathbf{d}}_\mathbf{i}$. However, experimental results suggested that the MSE loss gives robust direction predictions that are less variant across dataset scales.

Semantic segmentation loss. We use the standard cross-entropy loss to supervise the semantic segmentation stream, which is widely used for multi-class classification tasks, i.e.,

$$L_S = -\sum_{i=1}^{N} y_i^s \log p^s(\mathbf{x_i}), \tag{3.6}$$

where $y_i^s \in R^K$ denotes the one-hot vector of the GT semantic label s of the i^{th} point. $p^s(\mathbf{x_i})$ is the predicted probability of the i^{th} point for the GT category obtained from the network softmax layer.

Consequently, the total network loss is given by

$$L = L_S + \lambda_1 L_B + \lambda_2 L_D, \tag{3.7}$$

where we use λ_1 and λ_2 to balance between different losses.

3.4. EXPERIMENTS

3.4.1. DATASETS

We assess the effectiveness of our approach for the semantic segmentation of 3D point clouds across both indoor and outdoor urban scene datasets.

For indoor scenes, we use the challenging Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset, a large-scale point cloud dataset representing indoor areas of six buildings with diverse architectural styles (Armeni *et al.*, 2016). S3DIS covers 6020 square meters and includes a variety of spaces such as personal offices, conference rooms, exhibition areas, and hallways. Each point contains the xyz coordinates and color information. The dataset is labeled into 13 semantic classes: ceiling, floor, window, wall, beam, column, door, table, sofa, chair, bookcase, board, and clutter. In accordance with prior research Landrieu and Simonovsky, 2018; Qi, Su, *et al.*, 2017; Thomas *et al.*, 2019; H. Zhao *et al.*, 2021, we use Area 5 for testing and the rest areas for training.

For outdoor scenes, we evaluate using the SensatUrban dataset (Q. Hu *et al.*, 2021), a recent, large-scale UAV photogrammetry point cloud dataset spanning over 7.6 km^2. It consists of nearly three billion points collected from three cities in the UK. Each point is annotated with detailed semantic labels, including ground, vegetation, building, wall, bridge, parking, rail, traffic road, street furniture, footpath, car, bike, and water. Following the SensatUrban recommended data split, we use blocks 1 and 5 from Birmingham and blocks 7 and 10 from Cambridge for validation. Blocks 2 and 8 from Birmingham, and blocks 15, 16, 22, and 27 from Cambridge are used for testing. The remaining tiles are utilized for training.

Both GT boundary maps and direction maps are directly derived from the raw dataset. To generate GT boundary maps, we use kNN search for each point in the dataset. We empirically set $k = 4$ to ensure sharp object boundaries. A point is recognized as a boundary point if its semantic label differs from any neighboring labels. To generate GT direction maps, we identify the nearest boundary point for each point, and the direction is assigned as the vector pointing from the closest boundary point to the current point. The GT direction vectors are normalized with a magnitude of one and are then used for training.

3.4.2. IMPLEMENTATION DETAILS

We choose two state-of-the-art network architectures as our backbone: PointNet++ (Qi, Yi, *et al.*, 2017) and KP-Conv (Thomas *et al.*, 2019). Pointnet++ recursively applies point-wise MLP operators over a nested partitioning of the point clouds to extract hierarchical point features. In contrast, KP-Conv leverages point convolution within 3D Euclidean space. Both baseline networks employ a CNN-like architecture with the encoding-decoding strategy.

Built upon the baseline networks for point feature learning, we implement our method using Pytorch (Paszke *et al.*, 2019). Regarding the hyperparameters, we set $\alpha = 1.0$ and $r = 0.125$ for guided feature propagation as specified in Equation 3.2. For supervising the boundary stream in Equation 3.4, we set $\beta = 0.6$. The loss weight coefficients in Equation 3.7 are set to $\lambda_1 = 3.0$ and $\lambda_2 = 0.3$. To ensure a fair comparison, we maintain the same training settings as the baseline networks, such as the momentum for gradient descent optimization, base learning rate, and

learning rate schedule. Besides, we adopt unweighted cross-entropy loss for all the segmentation experiments.

Regarding data preprocessing, both the S3DIS and SensatUrban datasets consist of large 3D scenes that are too big for the baselines to process directly. For handling such data, PointNet++ randomly samples a fixed number of points (i.e., 4096) from each scene for network training. KP-Conv first applies grid sampling to reduce the number of input points. Then, it randomly samples 3D spheres within the scene to generate input batches for network training. During inference, 3D spheres are sampled regularly with a potential term to ensure that points can be visited by the network multiple times from different sphere locations. Finally, semantic predictions on grid-sampled clouds are mapped back to the original clouds. We follow the same data sampling strategies as the baseline networks for consistency. Our network training and data preprocessing details are illustrated in Table 3.1.

	Settings	PointNet++	KP-Conv	
		S3DIS	S3DIS	SensatUrban
Data	sampling strategy	random	grid + sphere	
	points per scene	4096	-	-
	grid size	-	5cm	20cm
	sphere radius	-	1.5m	9.0m
Training	base learning rate	0.001	0.01	0.01
	scheduler	70% per 10 epochs	98% per 1 epoch	
	momentum	-	0.98	0.98
	batch size	16	6	6
	steps per epoch	2973	300	600
	epochs	32	500	550

Table 3.1: Details of data preprocessing and network training.

3.4.3. EVALUATION METRICS

To quantitatively evaluate the performance of 3D semantic segmentation, we use the three standard metrics: overall accuracy, mean intersection over union, and per-category intersection over union scores (Lateef & Ruichek, 2019). Let K be the number of semantic categories, n_{ij} be the number of points that belong to class i that are classified into class j, and $N_i = \sum_{j=1}^{C} n_{ij}$ be the total number of points of class i, the three metrics are defined as following:

Overall Accuracy (OA). It measures the per-point accuracy of semantic labeling, which is simple to calculate and easy to use. However, a major limitation is that it ignores the class imbalance issue commonly present in large urban scenes.

$$OA = \frac{\sum_{i=1}^{K} n_{ii}}{\sum_{i=1}^{K} N_i} \tag{3.8}$$

Intersection over Union (IoU). It focuses on the amount of overlap between GT labels and semantic predictions. By calculating the ratio of the intersection size to the union size, IoU estimates how well the model distinguishes a particular class from the others.

$$IoU_i = \frac{n_{ii}}{N_i - n_{ii} + \sum_{j=1}^{K} n_{ji}} \tag{3.9}$$

mean Intersection over Union (mIoU). It calculates the average IoU scores across all classes. Compared to OA, mIoU provides a more informative evaluation and is less affected by class imbalance.

$$mIoU = \frac{1}{K} \sum_{i=1}^{K} IoU_i \tag{3.10}$$

3.4.4. RESULTS OF INDOOR SCENES

We evaluate our method on the indoor dataset S3DIS (Armeni *et al.*, 2016) using the two networks, PointNet++ (Qi, Yi, *et al.*, 2017; X. Yan, 2019) and KP-Conv (Thomas *et al.*, 2019), as our backbones. In Table 3.2, we report the performance comparison between these two baselines and our corresponding networks. These methods are either trained using only 3D coordinates and color information (w/o N), or using 3D coordinates, color information, and normals (w/ N). We consider incorporating normals in training, as they aid in distinguishing between boundary and object interior points. Notably, for the PointNet++ backbone, we report averaged scores over five training runs to account for performance variations during network training.

Compared to the baselines, our boundary-aware feature propagation mechanism consistently improves both OA and mIoU scores. When trained using only 3D coordinates and color information, our method achieves mIoU gains of 0.3% and 1.7% with the PointNet++ and KP-Conv backbones, respectively. When incorporating 3D coordinates, color information, and normals for training, our method yields mIoU gains of 1.2% and 1.6% with PointNet++ and KP-Conv, respectively. In particular, we observe significant improvements in categories such as *column*, *window*, *door*, and *board*. For these categories, the baseline methods often encounter boundary segmentation errors, whereas our approach achieves more precise boundary delineation. However, in certain scenarios, our method may also propagate segmentation errors to a larger extent. This leads to a decrease in mIoU scores for specific categories, such as *board*. Despite this, the overall quantitative results indicate that our approach generally enhances semantic segmentation performance across most categories.

Method	OA(%)	**mIoU(%)**	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet++ (w/o N)	83.5	53.6	89.6	**97.6**	74.6	0.0	4.7	55.1	**19.9**	78.2	68.9	**66.3**	41.7	55.6	**44.2**
+ Ours	**83.6**	**53.9**	**89.7**	97.2	**75.4**	0.0	**7.4**	**60.8**	14.1	**79.0**	**69.5**	65.8	**42.1**	**57.0**	42.6
PointNet++ (w/ N)	83.9	53.9	91.4	96.6	76.0	0.0	8.0	**53.7**	16.9	81.8	**70.5**	63.8	48.9	49.4	43.6
+ Ours	**84.3**	**55.1**	**91.5**	**97.2**	76.0	0.0	**13.8**	53.4	**19.1**	**83.6**	70.4	**65.4**	**49.6**	**50.3**	**45.4**
KP-Conv rigid (w/o N)	-	65.4	92.6	97.3	81.4	0.0	16.5	54.5	69.5	**80.2**	90.1	66.4	**74.6**	**63.7**	58.1
+ Ours	89.7	**67.1**	**94.0**	**97.9**	**82.6**	0.0	**23.3**	**56.6**	**75.4**	80.1	**91.1**	**75.7**	74.4	62.3	**59.1**
KP-Conv rigid (w/ N)	89.2	65.6	**94.0**	97.9	81.6	0.0	20.0	54.5	64.4	80.1	91.6	77.4	73.8	59.2	59.0
+ Ours	**89.6**	**67.2**	93.9	97.9	**82.7**	**0.2**	**23.8**	**55.0**	**73.7**	**80.5**	**91.8**	**77.7**	**74.1**	**63.2**	59.0

Table 3.2: Semantic segmentation results achieved on the S3DIS dataset (Armeni *et al.*, 2016), following the instructions of the officially released code of PointNet++ (Qi, Yi, *et al.*, 2017; X. Yan, 2019) and KP-Conv (Thomas *et al.*, 2019). These methods are trained using either 3D coordinates and color information (w/o N) or with additional normal information (w/ N). OA (%), mIoU (%), and per-category IoU scores are reported. We report averaged scores over five training runs for PointNet++ based networks to account for performance variations.

In Table 3.3, we present quantitative comparisons with other state-of-the-art methods on the S3DIS benchmark. While our approach does not surpass the performance of Point Transformer (H. Zhao *et al.*, 2021), it provides valuable insights into feature propagation mechanisms within the decoding process to enhance commonly used networks. Experimental results have shown the effectiveness of our proposed feature propagation strategy.

Method	OA(%)	**mIoU(%)**
PointNet (Qi, Su, *et al.*, 2017)	-	41.1
TangentConv (Tatarchenko *et al.*, 2018)	82.5	52.8
SPGraph (Landrieu & Simonovsky, 2018)	86.4	58.0
BGENet (Gong *et al.*, 2021)	-	61.4
RandLA-Net (Q. Hu *et al.*, 2020)	87.2	62.4
IAFNet (M. Xu, Zhou, *et al.*, 2021)	88.4	64.6
JSENet (Z. Hu *et al.*, 2020)	-	67.7
Point Transformer (H. Zhao *et al.*, 2021)	**90.8**	**70.4**
PointNet++ (Qi, Yi, *et al.*, 2017) (w/o N)	83.5	53.6
+ ours	83.6↑	53.9↑
PointNet++ (Qi, Yi, *et al.*, 2017) (w/ N)	83.9	53.9
+ ours	84.3↑	55.1↑
KP-Conv rigid (Thomas *et al.*, 2019) (w/o N)	-	65.4
+ ours	89.7	67.1↑
KP-Conv rigid (Thomas *et al.*, 2019) (w/o N)	89.2	65.6
+ ours	89.6↑	67.2↑

Table 3.3: Semantic segmentation performance comparison with other state-of-the-art works on the S3DIS dataset, accessed in September 2022. BGENet, IAFNet, and JSENet also consider local boundary refinement in their approach design. Our approach brings consistent performance improvements to baseline networks, as indicated by the up arrow ↑.

Figure 3.5 and Figure 3.6 present the segmentation results on the indoor dataset S3DIS, adopting KP-Conv and PointNet++ as the backbone, correspondingly. We use the identical color schemes to render the scenes in both figures: for *ceiling*, for *floor*, for *wall*, for *beam*, for *column*, for *window*, for *door*, for *table*, for *chair*, for *sofa*, for *bookcase*, for *board*, and for *clutter*.

Both qualitative comparisons demonstrate that our method consistently enhances semantic segmentation across different baseline architectures. In particular, the joint learning framework and the boundary-guided feature propagation strategy effectively mitigate segmentation errors near object boundaries for various categories, such as *window*, *column*, and *board*. Besides, our method also improves segmentation

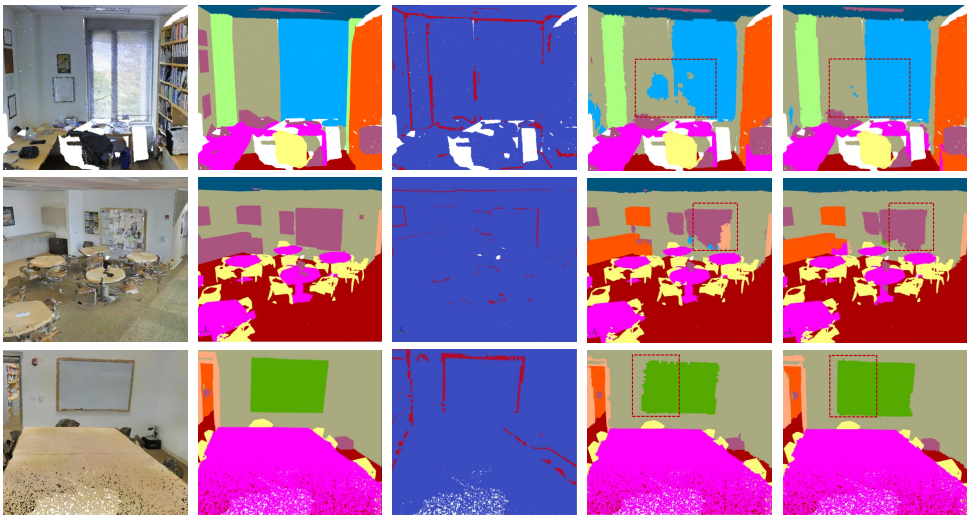


Figure 3.5: Qualitative results on the S3DIS dataset (Armeni *et al.*, 2016), using KP-Conv (Thomas *et al.*, 2019) as the backbone. The network is trained using 3D coordinates, colors, and normals. In the third column, the predicted boundary points are rendered in red.

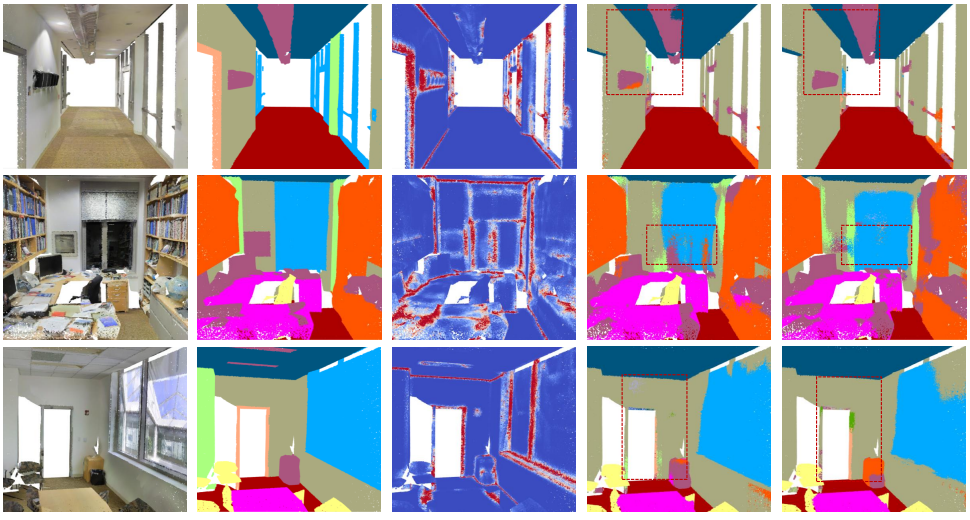


Figure 3.6: Qualitative results on the S3DIS dataset (Armeni *et al.*, 2016), using PointNet++ (Qi, Yi, *et al.*, 2017) as the backbone. The network is trained using 3D coordinates, colors, and normals. In the third column, the boundary probability maps are visualized: blue for low and red for high.

smoothness in local regions, especially within object interiors (Figure 3.6 row 2). Notably, even though the predicted boundaries may not perfectly align with the GT, they still provide informative guidance for feature propagation in the decoding layers, facilitating locally consistent segmentation near object boundaries.

3.4.5. RESULTS OF OUTDOOR SCENES

In addition to indoor scenes, we evaluated the performance of our method on the outdoor SensatUrban dataset (Q. Hu *et al.*, 2021), employing KP-Conv (Thomas *et al.*, 2019) as the backbone network. Table 3.4 presents the performance of our approach in comparison with KP-Conv and other state-of-the-art methods.

Our method has achieved an OA of 93.9% and an mIoU of 59.7%, surpassing all major competing approaches on this benchmark as of September 2022. The results demonstrate the capability of our approach in generalizing to large-scale outdoor urban scenes. Compared to the state-of-the-art baseline KP-Conv Thomas *et al.*, 2019, our boundary-guided feature propagation strategy improves OA by 0.6% and mIoU by a margin of 2.1%. In terms of per-category performance, improvements in most categories can also be observed, particularly in those categories characterized by distinct geometric boundaries, such as *bridges*, *traffic roads*, and *footpaths*. Our method achieves IoU gains of 21.0%, 6.1%, and 1.6% for *bridges*, *traffic roads*, and *footpaths*, respectively. However, due to the class imbalance in the SensatUrban dataset, minority categories such as *rail* and *bike* remain challenging to recognize for most methods listed in the table, including our approach. Besides, compared to the baseline, our method exhibits a performance decline of 20.8% IoU in the *water* category. This decline can be attributed to the irregular geometric boundaries of water in urban environments, which introduce noise in both boundary detection and direction prediction, ultimately yielding less accurate segmentation.

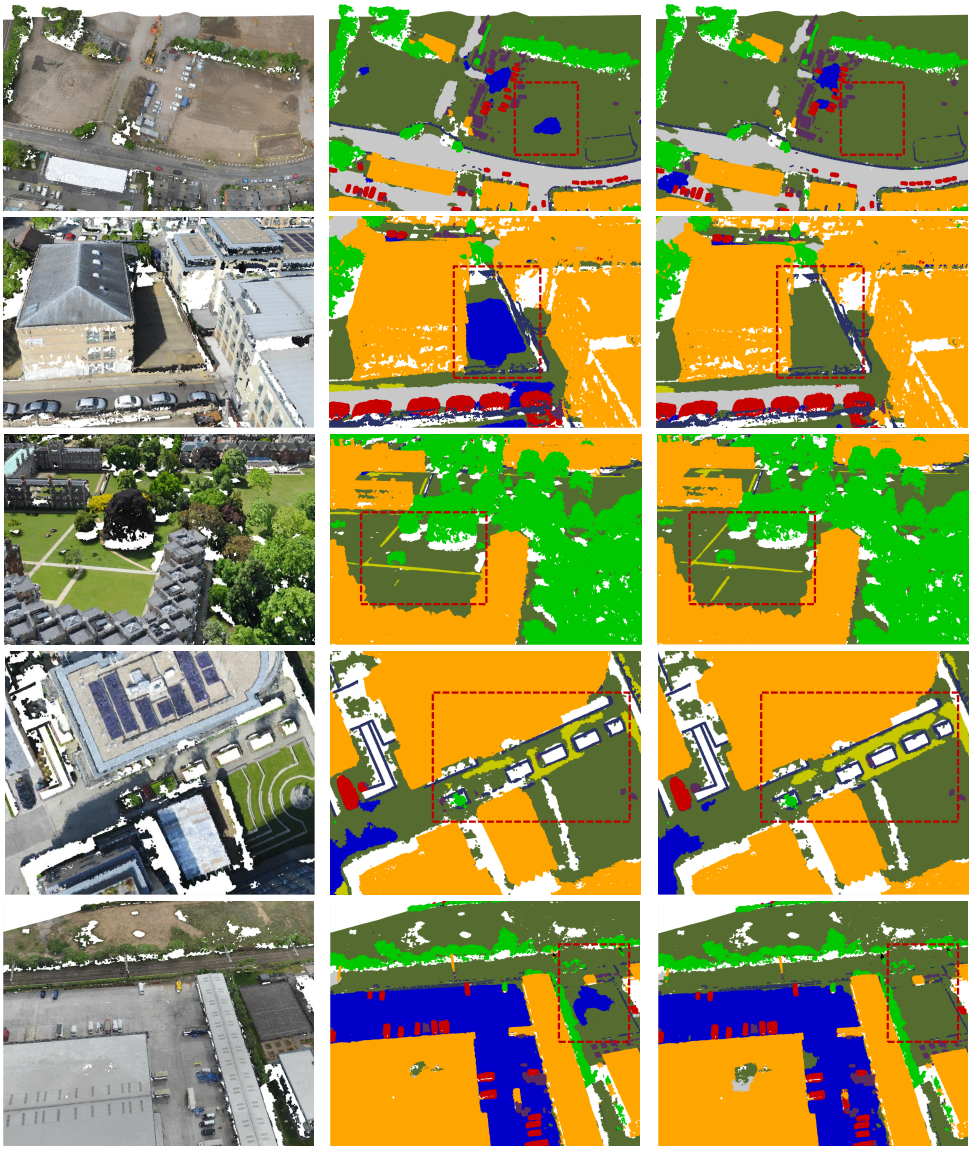
Figure 3.7 presents the segmentation results on the SensatUrban dataset. GT labels are not included, as the true labels for the test set are not publicly available. Despite the absence of GT visualization, comparisons with baseline results still indicate that our method achieves superior segmentation performance. Specifically, our approach improves the localization of object boundaries for minor categories, such as *parking* and *footpaths*. Meanwhile, our method enhances local segmentation consistency by reducing segmentation noise within object interiors.

3.4.6. ABLATION STUDIES

In this section, we present ablation studies to support the contributions of our methodology design. First, we assess the effectiveness of each individual module by systematically removing it from the network and evaluating the impact on segmentation performance. Next, as our key contribution lies in the design of guided feature propagation for local boundary refinement, we further investigate the effectiveness of this propagation mechanism applied across various decoding layers.

Method	OA(%)	**mIoU(%)**	ground	veg.	building	wall	bridge	parking	rail	traffic.	street.	car	footpath	bike	water
PointNet (Qi, Su, *et al.*, 2017)	80.8	23.7	68.0	89.5	80.1	0.0	0.0	4.0	0.0	32.0	0.0	35.1	0.0	0.0	0.0
PointNet++ (Qi, Yi, *et al.*, 2017)	84.3	32.9	72.5	94.2	84.8	2.7	2.1	25.8	0.0	31.5	11.4	38.8	7.1	0.0	56.9
TangentConv (Tatarchenko *et al.*, 2018)	77.0	33.3	71.5	91.4	75.9	35.2	0.0	45.3	0.0	26.7	19.2	67.6	0.0	0.0	0.0
SPGraph (Landrieu & Simonovsky, 2018)	85.3	37.3	69.9	94.6	88.9	32.8	12.6	15.8	**15.5**	30.6	23.0	56.4	0.5	0.0	44.2
SparseConv (Graham *et al.*, 2018)	88.7	42.7	74.1	97.9	94.2	63.3	7.5	24.2	0.0	30.1	34.0	74.4	0.0	0.0	54.8
RandlaNet (Q. Hu *et al.*, 2020)	89.8	52.7	80.1	98.1	91.6	49.0	40.8	51.6	0.0	56.7	33.2	80.1	32.6	0.0	71.3
KP-Conv (Thomas *et al.*, 2019)	93.2	57.6	**87.1**	**98.9**	95.3	74.4	28.7	41.4	0.0	56.0	54.4	85.7	40.4	0.0	**86.3**
Ours	**93.8**	**59.7**	85.8	98.9	**96.8**	**79.3**	**49.7**	**52.4**	0.0	**62.1**	**57.5**	**86.8**	**42.0**	0.0	65.5

Table 3.4: Semantic segmentation results on the SensatUban dataset Q. Hu *et al.*, 2021, evaluated using the Birmingham block 2, 8, and the Cambridge block 15, 16, 22, and 27. OA (%), mIoU (%), and per-category IoU scores are reported. The results of the seven competing networks are from the original SensatUrban benchmark paper Q. Hu *et al.*, 2021 as accessed in September 2022. To ensure a fair comparison with these networks, we train our method using only 3D coordinates and color information.



Input point clouds Baseline results Our results

Figure 3.7: Qualitative results on the SensatUrban dataset (Q. Hu *et al.*, 2021), using KP-Conv (Thomas *et al.*, 2019) as the backbone. The network is trained using 3D coordinates and colors. We do not visualize the GT segmentation as the GT labels for the test set are not accessible. We use the following colors to render distinct urban objects: *ground*, *vegetation*, *building*, *wall*, *bridge*, *parking*, *rail*, *traffic road*, *street furniture*, *car*, *footpath*, *bike*, and *water*.

The S3DIS dataset is used for ablation experiments, as it provides access to GT semantic labels for all points. We conduct our ablation studies using KP-Conv (Thomas *et al.*, 2019) as the backbone network. Following previous works (Landrieu & Simonovsky, 2018; Qi, Su, *et al.*, 2017; Thomas *et al.*, 2019; H. Zhao *et al.*, 2021), we use Area 5 for testing and the remaining areas for training. All experiments are conducted under the same hyperparameter settings, as detailed in Table 3.1.

NETWORK COMPONENT

Our method leverages a multi-task learning framework in conjunction with guided feature propagation to improve 3D semantic segmentation, particularly in local boundary regions. To evaluate the contribution of each network component, we conduct ablation studies by systematically removing individual components and assessing their impact on performance. The results of these experiments are summarized in Table 3.5.

Method	OA(%)	**mIoU(%)**
(1) Baseline network	89.2	65.6
(2) Baseline + Boundary Stream	89.3	66.6
(3) Baseline + Direction Stream	89.1	66.2
(4) Full network with SFP	89.4	66.3
(5) Full network with GFP	**89.6**	**67.2**

Table 3.5: mIoU scores of the ablated networks from (1) to (5). SFP denotes standard feature propagation, and GFP denotes guided feature propagation. All networks are trained using 3D coordinates, color information, and normals.

From Table 3.5, we observe that incorporating the boundary stream into the baseline network improves the mIoU by 1.0% and OA by 0.1%. Adding the direction stream results in a 0.8% increase in mIoU, leading to a slight reduction of 0.1% in OA. When both streams are incorporated with standard feature propagation, the network achieves a 0.2% gain in OA and a 0.7% gain in mIoU compared to the baseline. Notably, combining both the boundary and direction streams leads to a 0.3% reduction in mIoU compared to using only the boundary stream. Nevertheless, the direction stream still provides valuable information for subsequent segmentation refinement. Last, employing the proposed boundary-aware feature propagation mechanism, we observe an additional performance boost, with OA increasing by 0.4% and mIoU improving by 1.6%. These findings reveal that the proposed method effectively enhances feature inheritance, leading to more purified and discriminative feature representations.

GUIDED FEATURE PROPAGATION

As guided feature propagation plays a key role in our methodology design, we further analyze the effectiveness of the proposed feature propagation mechanism when applied to different decoding layers.

In network decoding, feature maps undergo four upsampling stages, sequentially progressing from the fifth layer to the fourth, from the fourth to the third, from the third to the second, and so forth (as illustrated in Figure 3.4). To assess the impact on network performance, we replace Standard Feature Propagation (SFP) with the proposed Guided Feature Propagation (GFP) at each upsampling stage.

We propagate features of neighboring points $\mathbf{x_j}$ in the l^{th} layer to points $\mathbf{x_i}$ in the $l-1^{th}$ layer. For GFP, the feature propagation process is governed by the weighting functions introduced in Section 3.3.2, as defined by Equation 3.2. For SFP, we adopt the inverse distance weighting strategy following PointNet++ (Qi, Yi, *et al.*, 2017) and KP-Conv (Thomas *et al.*, 2019), which operates within a local neighborhood and is formulated as follows:

$$w(\mathbf{x_j}) = \frac{1}{||\mathbf{x_j} - \mathbf{x_i}||_2 + \epsilon}, \tag{3.11}$$

where ϵ represents a small threshold that prevents the denominator from being zero.

For SFP, we use a fixed radius to search for the local neighbors since it is more robust to varying point densities compared to kNN. The base radius in the first layer is set to 6.25cm and doubles at each subsequent layer. For GFP, as the predicted directions and boundaries naturally help to mitigate the influence of outliers, we still use kNN. We empirically set $k = 8$.

GFP layers	SFP layers	OA(%)	**mIoU(%)**
-	up1, up2, up3, up4	89.4	66.3
up1	up2, up3, up4	**89.6**	**67.2**
up1, up2	up3, up4	89.7	66.9
up1, up2, up3	up4	89.6	66.6
up1, up2, up3, up4	-	89.4	66.0

Table 3.6: OA and mIoU scores of the ablated networks adopting GFP in various decoding layers. SFP: standard feature propagation. GFP: our guided feature propagation. The experiments are conducted on the S3DIS dataset (Armeni *et al.*, 2016) using KP-Conv (Thomas *et al.*, 2019) as the backbone, trained using 3D coordinates, color information, and normals.

Table 3.6 presents the segmentation performance when applying GFP at different feature upsampling layers. The highest mIoU score, as previously reported in Section 3.4.4, is achieved when GFP is used exclusively in the first upsampling

layer while SFP is applied in the rest layers. When replacing SFP with GFP in higher layers, we observe a slight drop in the performance. However, replacing SFP with GFP in higher upsampling layers results in a slight performance decline, with the lowest performance observed when GFP is applied across all upsampling stages. We attribute this trend to the design of GFP, which primarily focuses on recovering local boundary details. Therefore, its effectiveness is most highlighted in lower-level upsampling layers. Conversely, in higher-level upsampling layers, SFP is more advantageous since it better preserves global contextual features during feature propagation.

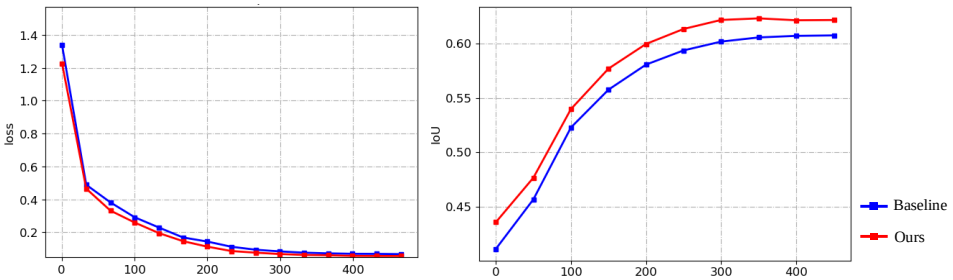
3.4.7. COMPLEXITY AND CONVERGENCE ANALYSIS

In this section, we compare the convergence and efficiency of our network with the baseline methods. The corresponding results are reported in Table 3.7.

Our network contains a significantly larger number of parameters than the baselines, as it employs three distinct decoders to handle the three downstream tasks. Consequently, it leads to increased computational complexity, resulting in lower speed in training and inference. Compared to PointNet++, our method increases training time by 23% and inference time by 15%. Compared to KP-Conv, we observe an average increase of 55% in training time and 33% in inference time. However, even with the increased computational complexity, our network exhibits faster convergence. This is demonstrated in Figure 3.8, where we compare the segmentation loss and validation curves of our network against the baseline. The results indicate that our network reaches convergence at approximately 300 epochs, whereas the baseline requires around 400 epochs to converge, highlighting the efficiency of our approach in optimizing the learning process.

		PointNet++	KP-Conv	
		S3DIS	S3DIS	SensatUrban
#Params	Baseline	0.97	24.38	24.38
(M)	Ours	2.19	32.78	32.78
Training time	Baseline	0.52	0.09	0.13
(sec./batch)	Ours	0.64	0.14	0.20
Inference time	Baseline	0.71	0.05	0.08
(sec./batch)	Ours	0.82	0.07	0.10

Table 3.7: Comparison of our running time against the baseline, e.g., PointNet++ (Qi, Yi, *et al.*, 2017) and KP-Conv (Thomas *et al.*, 2019), on the S3DIS (Armeni *et al.*, 2016) and SensatUrban (Q. Hu *et al.*, 2021) datasets. The total number of learnable parameters is an indicator of model complexity. We use the average running time per batch as the efficiency indicator. All experiments are conducted with an NVIDIA RTX2080Ti GPU.



(a) Segmentation losses (b) Validation curves

Figure 3.8: Comparison of network convergence. The horizontal axes denote the number of training epochs. The statistics are obtained on the S3DIS dataset (Armeni *et al.*, 2016) using KP-Conv (Thomas *et al.*, 2018) as the backbone, trained using 3D coordinates, color information, and normals.

3.4.8. LIMITATIONS

Our proposed method performs boundary prediction, direction prediction, and semantic segmentation in parallel to enhance the accuracy of local semantic segmentation. By adopting a boundary-aware guiding mechanism for feature propagation, our approach generates sharper and more purified feature maps that effectively reduce segmentation errors in boundary regions. However, despite these advantages, the method still has certain limitations.

First, the designed feature propagation mechanism is performed within local neighborhoods. This enhances feature consistency within an object's interior at a local scale, while failing to explicitly model long-range dependencies, thus limiting its ability to guarantee global feature consistency due to the lack of global contextual relationship reasoning. Second, since our work focuses on guiding features to recover boundary information in the decoding layers, it naturally cannot cope with potential information loss in the feature encoding stages. Last, our approach requires extra preprocessing steps. We need to generate GT boundary maps and direction maps for training, which increases computational overhead and memory consumption.

3.5. CONCLUSION

In this chapter, we introduce a novel boundary-guided feature propagation mechanism to improve the semantic segmentation of 3D point clouds, aiming to push and guide the object boundaries towards the desired locations. Our network jointly learns boundary maps, direction maps, and point-wise semantic labels in an end-to-end manner. Extensive studies on the S3DIS and SensatUrban datasets have demonstrated the effectiveness of our approach. Our experiments and analysis reveal two key factors contributing to the improvements in segmentation performance. First, the joint learning of the three tasks mutually enhances the shared

feature encoder. Second, the predicted boundaries and directions serve as effective guides, allowing points to inherit features from more homogeneous regions, which compensates for the loss of local boundary information in the network decoding layers. Our approach is particularly advantageous for categories with clear geometric boundaries, such as doors, windows, and street paths.

However, as discussed in Section 3.4.8, the improvements achieved by our method are still limited. The primary limitation arises from the locally performed feature propagation, which cannot optimize segmentation results on a global level. Besides, since the three downstream tasks have different levels of complexity, combining them into a single network limits the capability of the shared feature encoder to learn discriminative features for all three tasks. In future work, we would like to explore the adaptive boundary detection method (M. Xu, Zhou, *et al.*, 2021) to achieve higher-quality delineation of semantic boundaries. We would also like to extend our boundary-aware feature propagation mechanism by incorporating global contextual reasoning techniques to further enhance 3D semantic segmentation performance.

4

GLOBAL PROTOTYPE EXPANSION FOR SEMANTIC SEGMENTATION

Chapter 3 investigates 3D semantic segmentation from a local perspective. While the proposed local boundary-guided approach effectively mitigates segmentation errors near object boundaries, it lacks the capacity for global contextual reasoning in complex urban scenes. Furthermore, most existing semantic segmentation methods also rely on local feature aggregators with limited receptive fields, which poses a bottleneck for their performance. To address these limitations, this chapter presents a novel computational block, Prototype Expansion, designed to facilitate efficient global analysis using class prototypes. We explicitly allow individual points to interact with prototypes generated across batches to achieve globally consistent classification. The proposed PE module consists of two integral parts: a prototype generator that constructs a prototype vector for each class in a non-parametric manner, and an Expansion operator projecting prototypes to point embeddings based on their feature-wise dependencies. In particular, the Expansion operator is inspired by the Attention mechanism but achieves more effective feature recalibration with fewer parameters. Integrating the PE module into four prominent point cloud learning networks consistently demonstrated performance boosts on both indoor and outdoor datasets. These experiments demonstrate the general applicability of PE, showing its effectiveness as a booster for a wide range of semantic segmentation tasks.

4.1. INTRODUCTION

With the rapid development of LiDAR sensors, the interpretation and modeling of urban scenes have progressively transitioned from 2D to 3D representations. Specifically, 3D point clouds with enriched semantics are fundamental for numerous downstream applications. They provide valuable inputs for analyzing urban layouts and land cover types, as well as for the accurate reconstruction of 3D urban structures (J. Huang *et al.*, 2022). Moreover, the integration of segmented point cloud data with external data sources such as infrastructure maps facilitates navigation (Anthes *et al.*, 2016) and autonomous driving (Yurtsever *et al.*, 2020). Nevertheless, as detailed in Chapter 2, point clouds still pose significant challenges for deep learning due to their sparsity, irregularity, and lack of explicit topological structure. These characteristics hinder the direct application of traditional deep neural network architectures, such as CNNs, to 3D semantic segmentation tasks.

The pioneering PointNet (Qi, Su, *et al.*, 2017) and PointNet++ (Qi, Yi, *et al.*, 2017) made it possible to directly process raw point clouds using deep neural networks. Since then, many point-wise networks have been proposed, with a specific focus on the design of local feature aggregators (Q. Hu *et al.*, 2020; H. Lin *et al.*, 2023; Qian *et al.*, 2022; Thomas *et al.*, 2019). In particular, recent works such as (H. Lin *et al.*, 2023; Qian *et al.*, 2022) enhance the original PointNet++ by incorporating model scaling and improved training strategies. On the other hand, transformers, originating from NLP (Vaswani *et al.*, 2017), have also shown great potential in 3D vision tasks recently. By adopting the attention mechanism, transformers can model the relationships of tokens in a set, which makes them particularly well-suited for point cloud segmentation tasks since points can be viewed as a discrete set in the 3D space. Inspired by this observation, several studies have investigated transformer-based architectures to process point clouds (Lai *et al.*, 2022; X. Wu *et al.*, 2022; C. Zhang *et al.*, 2022; H. Zhao *et al.*, 2021). Among them, Point Transformer and its successor (X. Wu *et al.*, 2022; H. Zhao *et al.*, 2021) apply local self-attention within kNN regions, which may constrain the receptive field. To address this limitation, Stratified Transformer (Lai *et al.*, 2022) expands the receptive field by incorporating far-distant, stratified key point sampling.

While these methods have achieved remarkable performance on various 3D semantic segmentation tasks, they suffer from two major drawbacks: First, the receptive fields are limited since the network operators are often designed to be local. Performing global operators such as global attention is possible, however can be costly in terms of GPU and memory resources. Second, the network supervision using the cross-entropy loss only considers between-class discrepancy and ignores intra-class compactness. This can potentially lead to feature-wise variations among individual points within the same class, thus downgrading the performance of downstream tasks.

To address the above-mentioned problems, this chapter explores semantic segmentation from the global prototype perspective. Prototypes, dating back to cognitive models (Rosch, 1973), aim to describe each class using the most

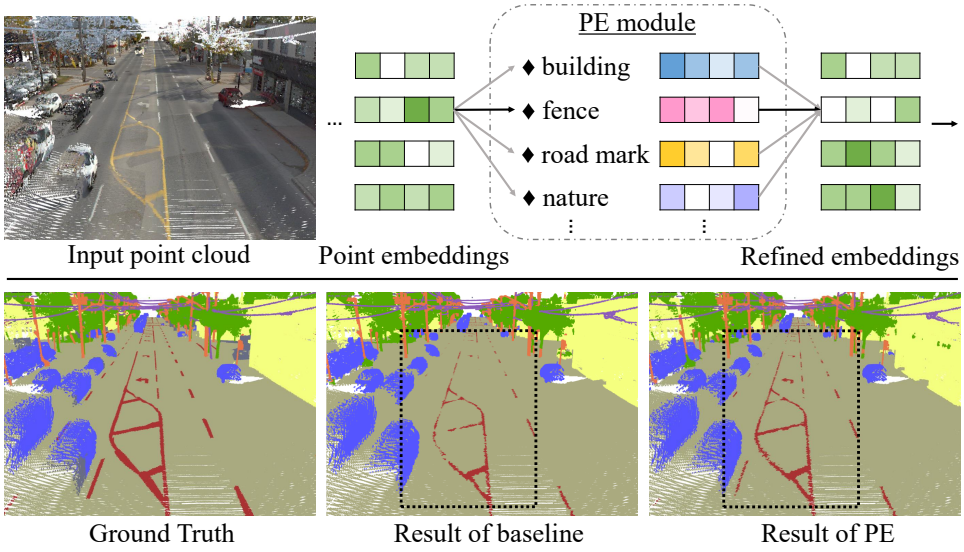


Figure 4.1: Our PE module allows for point-to-prototype interactions to enhance the individual point embeddings (top). Without additional supervision loss terms, PE consistently improves the overall scene segmentation performance, especially for minor object categories such as road marks (bottom). The visual comparison was generated on the Toronto3D dataset (W. Tan *et al.*, 2020).

representative samples, and they thus naturally capture global information. Furthermore, the proximity of other data samples to the prototypes gives a good reflection of intra-class compactness. Although prototypes have been investigated in a few studies (J. Li & Dong, 2023; M. Liu *et al.*, 2022; T. Zhou *et al.*, 2022), they have only served as additional loss constraints or as a non-parametric scheme for network inference. Their potential to enhance point cloud feature representation has not been fully exploited yet.

We introduce a novel network block, namely the *Prototype Expansion* (PE) module, which leverages prototypes for efficient global-level point cloud analysis, as illustrated in Figure 4.1. Prototypes are representative embeddings that describe the general characteristics of a specific class. We use a single prototype constructed from training point samples to abstract an object class. Our goal is to enhance the point feature representation by enabling each point to interact with the class prototypes. To achieve this, the proposed PE module captures the dependencies between points and prototypes, according to which the prototype vectors are then explicitly projected back to individual point embeddings. Such an operation explicitly incorporates global semantic information into individual point representations while maintaining computational efficiency.

The PE-based approach has two components. First, we obtain the per-class

prototype vectors in a non-learnable manner following T. Zhou *et al.* (2022). Since no learnable parameters are involved, this approach does not increase network complexity, which is beneficial for generalizability. Second, we perform a novel *Expansion* operation to expand the prototypes to point-level embeddings according to their dependencies in the feature space. For modeling such point-to-prototype dependencies, we simply use the feature inner product inspired by the attention mechanism (Vaswani *et al.*, 2017). Nevertheless, compared to standard attention, our proposed *Expansion* operator involves fewer network parameters and allows points to efficiently inherit information from highly related prototypes. In the end, a gating mechanism is utilized to effectively recalibrate individual point features concerning corresponding prototypes. By expanding and projecting global class prototypes to individual point embeddings, our PE module becomes a global operator. Meanwhile, the intra-class compactness of the learned feature representation is also increased, without imposing any extra loss terms. Our proposed PE module is computationally efficient since it only requires each point to interact with a small set of prototypes instead of the entire point set. With a slight increase in network complexity, we can enhance the point feature representation and thus improve the overall semantic segmentation performance.

Our contributions can be summarized in two folds:

- We propose the use of prototypes to build feature-wise connections between individual points and class descriptors for global analysis, as an efficient strategy to overcome the problem of the limited receptive fields.
- Following this insight, we design the PE module, which comes with a novel *Expansion* operator, to enhance the feature representation through point-to-prototype interactions. PE is flexible, easy to plug into existing networks, and consistently boosts the performance of 3D semantic segmentation tasks.

4.2. RELATED WORK

While there is a large volume of research on point cloud semantic segmentation, as reviewed in Section 2.2, most of these methods primarily emphasize local feature aggregation functions with limited receptive fields. In this section, we focus on methods explicitly designed to expand receptive fields and capture long-range contextual dependencies. As our approach design is partially inspired by transformers and self-attention mechanisms, we specifically examine recent advancements in 3D transformers for global point cloud analysis. In addition, we review relevant literature on prototype learning methodologies.

4.2.1. 3D TRANSFORMERS AND ATTENTION

Transformers were initially developed for NLP tasks (Vaswani *et al.*, 2017) and later also demonstrated strong performance in 2D image analysis (Z. Liu *et al.*, 2021).

Following their successful practice, recent approaches have extended transformers to 3D data, employing attention mechanisms for point cloud processing. The main motivation is that transformer architectures are inherently suited for point cloud data: the self-attention mechanism, which forms the foundation of transformer models, is by nature a set operator and well respects the permutation and cardinality invariance of the input points. Empowered by the self-attention mechanism, transformer models allow each point to selectively attend to other points based on their feature similarity, thereby modeling complex contextual relationships within the scene.

As reviewed in Section 2.2.3, a variety of transformer-based architectures have been proposed for point cloud semantic segmentation (M.-H. Guo *et al.*, 2021; P.-S. Wang, 2023; X. Wu *et al.*, 2022, 2024; H. Zhao *et al.*, 2021). However, many of these methods restrict the self-attention operation to local neighborhoods. Therefore, they are inherently limited by their local receptive fields and lack the capacity to model long-range dependencies. To address this, Stratified Transformer (Lai *et al.*, 2022) expands the receptive field by sampling sparse distant points as keys, which improves long-range contextual modeling. Alternatively, methods such as PatchFormer (C. Zhang *et al.*, 2022) and Point-TnT (Berg *et al.*, 2022) first segment the point cloud into 3D patches and then apply patch-based attention to capture global contextual information. Nevertheless, these patch-based strategies often require extensive preprocessing for global-level analysis.

4.2.2. PROTOTYPE LEARNING

The term prototype, referring to the representation interpretation of classes, dates back to cognitive science models (Rosch, 1973). It describes each semantic class using only one or a few best samples, while the class memberships for other samples are determined by their degree of similarity to the prototypes (Lakoff, 2007). Prototypes are also closely related to nearest neighbor classification (Cover & Hart, 1967) and clustering (Caron *et al.*, 2020; YM. *et al.*, 2020).

Recently, prototype learning has been increasingly applied to both 2D and 3D computer vision tasks. For instance, ProtoSeg (T. Zhou *et al.*, 2022) and Point-NN (R. Zhang *et al.*, 2023) use non-learnable prototypes for semantic inference, where prototype embeddings are initialized and progressively updated during training via momentum-based mechanisms. Semantic labels for pixels or points are then inferred based on their proximity to the nearest prototype in the feature space. Contrary to them, Y. Zhao *et al.* (2022) obtained prototypes via a parametric decoder and followed Maskformer (Cheng *et al.*, 2021) for the semantic inference. This strategy is computationally expensive due to the need for additional network layers to learn discriminative prototype embeddings. Furthermore, prototypes can also be used as soft constraints for 3D semantic segmentation tasks. M. Liu *et al.* (2022) employed a contrastive loss term that implicitly encourages feature embeddings of points to be more closely aligned with their corresponding class prototypes.

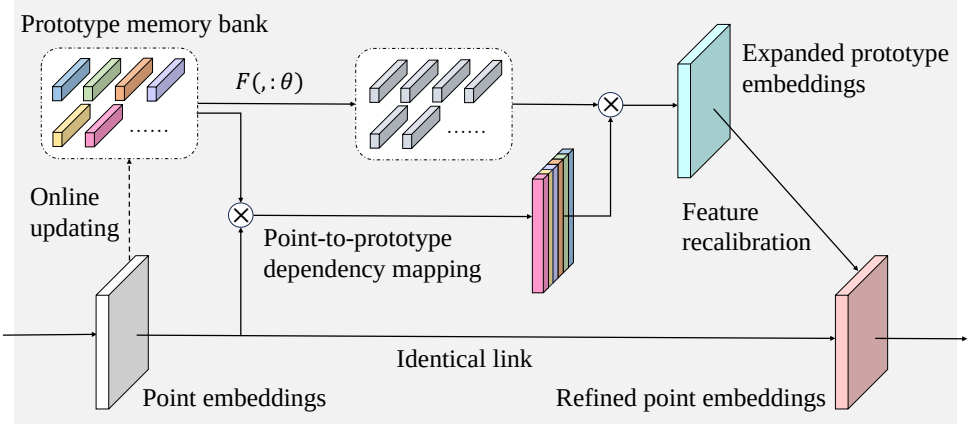


Figure 4.2: An overview of the *Prototype Expansion* module. $F(,:\theta)$ denotes a learnable function which is parameterized by θ. The height of the embedding maps represents the input point count, and the width represents the feature dimensionality.

Unlike these prior approaches, our method explicitly models the relationships between individual points and class prototypes, enabling the network to generate more discriminative point embeddings through an effective *Prototype Expansion* mechanism.

4.3. METHOD

Our proposed *Prototype Expansion* module is a computational module that takes point embeddings at an arbitrary level as input and refines them using global class prototypes. Figure 4.2 gives an overview of the PE module, which contains two main computational steps. First, the per-class prototypes are constructed and dynamically updated in a non-parametric manner, following the work of T. Zhou *et al.* (2022). Second, a novel *Expansion* operation is performed to expand the prototypes to point-level embeddings according to their dependencies in the feature space. Through this process, the PE module enhances the individual point feature representations learned by a network through global contextual information embedding.

4.3.1. PROTOTYPE CONSTRUCTION AND UPDATING

Efficient global analysis of point clouds is a challenging task. Due to the inherent complexity of 3D scenes and the varying data densities, it can have high GPU and memory demands. To achieve global analysis, previous studies (Berg *et al.*, 2022; C. Zhang *et al.*, 2022) have employed carefully designed data sampling strategies or

preprocessing techniques such as patch segmentation. However, these techniques often introduce additional computational complexity. To mitigate this problem, we propose to abstract the global information contained in a given scene using only a small set of class prototypes. Specifically, each class is represented by a single prototype embedding, thereby capturing the global context effectively without requiring additional data preprocessing steps.

Given a point cloud feature map $X \in R^{N \times F}$ with N the number of points and F the feature dimension, we can compress global information into a set with a limited number of prototypes $P \in R^{C \times F}$, where C is the number of classes and $C \ll N$. Here, the feature map X can originate from any arbitrary stage within the network. Similar to the previous studies of M. Liu *et al.* (2022) and T. Zhou *et al.* (2022), we construct the class prototypes in a non-parametric manner. Formally, the prototype embedding $\mathbf{p}_c \in R^F$ of a class c is initialized as a zero tensor. After each training iteration, this embedding is updated directly using the available training point samples, as described by the following equation:

$$\mathbf{p}_c \leftarrow \delta \mathbf{p}_c + (1-\delta)\frac{1}{N_c}\sum_{y_i=c}\mathbf{x}_i, \tag{4.1}$$

where $\mathbf{x}_i \in R^F$ denotes the feature embedding of the point i, $\delta \in [0,1]$ is the user-specified momentum coefficient, y_i is the point label, and N_c represents the total number of points belonging to class c in the current iteration.

It should be noted that prototypes can also be obtained through learnable network modules (Caron *et al.*, 2020; Y. Zhao *et al.*, 2022). However, in this study, we choose to adopt a non-parametric momentum updating mechanism to construct the class prototype embeddings. This design choice is motivated by two primary considerations. First, with fewer learnable parameters involved, we can reduce the network complexity, which can enhance generalization performance. In addition, since we abstract the prototypes directly from training point samples, we explicitly embed global contextual information into the prototype representations across training batches, allowing us to perform efficient global analysis.

4.3.2. PROTOTYPE EXPANSION

Having constructed the class prototype embeddings from training point samples, we introduce a novel *Expansion* operation to enhance individual point feature representation using the obtained prototypes. This expansion step projects and expands the prototypes back to individual point embeddings according to their dependencies in the feature space. By facilitating the interaction between individual point embeddings and global class descriptors, the expansion operation enables our PE module to function as a global operator. Furthermore, it allows each point to inherit enriched semantic information from the most relevant prototypes, which thereby naturally encourages intra-class compactness.

To achieve a global operator with enhanced intra-class compactness, the expansion function needs to be designed to satisfy two key criteria. First, the expansion operator should learn to capture the point-to-prototype dependencies, and accordingly, project only relevant prototypes to the point embeddings. Second, the operator must be formulated as a continuous and differentiable function to be compatible with backpropagation during training. Motivated by the design principle of transformers and self-attention (Vaswani *et al.*, 2017), we propose an attention-like function for *Prototype Expansion* to fulfill these criteria. Figure 4.3 details the computation scheme of the expansion operator.

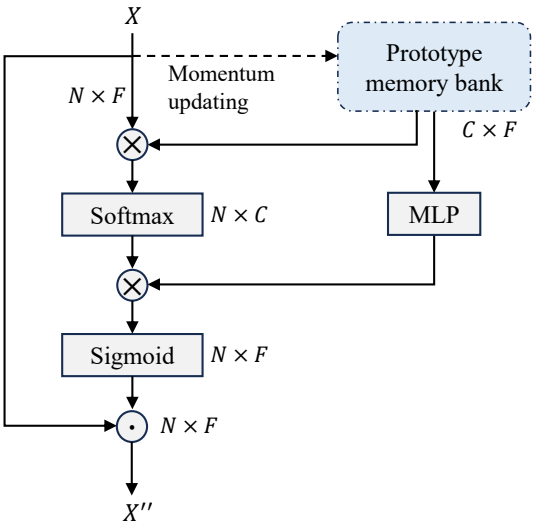


Figure 4.3: The computational graph of the *Expansion* operator. "×" denotes the feature inner product. In our PE module, only the MLP layer is involved with learnable parameters, making our module efficient in terms of both computation and memory.

First, we use the feature inner product to model the interdependencies between prototypes and individual point embeddings. The relation coefficient r of the point i and a prototype c is computed as follows:

$$r(\mathbf{x}_i, \mathbf{p}_c) = \mathbf{x}_i^T \mathbf{p}_c, \tag{4.2}$$

where $\mathbf{x}_i \in R^F$ is the i^{th} point embedding, and $\mathbf{p}_c \in R^F$ represents the obtained prototype of the class c.

The standard attention mechanism uses additional feature transformations such as learnable key $k(\cdot)$ and query $q(\cdot)$ functions to model the relation r. For instance, $r(\mathbf{x}_i, \mathbf{p}_c) = k(\mathbf{x}_i)^T q(\mathbf{p}_c)$. However, in our experiments, we observed that such learnable keys and queries resulted in slower network convergence and suboptimal

performance. One possible explanation is that the raw feature embeddings are already sufficient to capture the feature-wise dependencies, while adding additional learnable functions introduces unnecessary complexity, which can pose greater difficulties for network learning. Our prototype expansion, therefore, simply opts for the inner product between raw feature embeddings to model the relation r.

Having obtained the relation coefficient $r(\mathbf{x}_i, \mathbf{p}_c)$, we normalize it across all prototypes by Softmax. Subsequently, the prototypes are expanded to the individual point embeddings according to the following equation:

$$\mathbf{x}_i^{'} = \frac{\sum_{c=1}^{C} e^{r(\mathbf{x}_i, \mathbf{p}_c)} h(\mathbf{p}_c)}{\sum_{c=1}^{C} e^{r(\mathbf{x}_i, \mathbf{p}_c)}}, \tag{4.3}$$

where $h(\cdot)$ is a projection layer implemented as a linear layer, and C denotes the total number of classes. The obtained embedding map $\mathbf{x}_i^{'}$ essentially represents a compositional combination of prototypes per point, where closely related prototype vectors impose greater influences through their computed feature dependencies.

Last, we recalibrate the individual point embedding $\mathbf{x}_i$ using $\mathbf{x}_i^{'}$ with a gating mechanism, which is described as follows:

$$\mathbf{x}_i^{''} \leftarrow \mathbf{x}_i \odot \sigma(\mathbf{x}_i^{'}), \tag{4.4}$$

where $\sigma(\cdot)$ refers to the Sigmoid function and $\odot$ represents the Hadamard product. In Section 4.4.5, we provide ablation studies to demonstrate the effectiveness of this feature recalibration design strategy.

4.3.3. ANALYSIS

In this section, we present a detailed analysis of the proposed PE module, showing that our method effectively reduces computational complexity compared to other global analysis approaches on point clouds. Furthermore, we discuss the connections and distinctions between our method and existing studies in contrastive prototype learning.

Computational complexity. Our proposed PE module is computationally efficient compared to prior methods. This is mainly attributed to three reasons. First, we involve a minimal number of learnable parameters in both prototype construction and point-to-prototype dependency computation, which facilitates network optimization and thereby enables faster convergence. In addition, our proposed *Expansion* step functions as a global operator with linear computational complexity. Standard global attention (Vaswani *et al.*, 2017) has a quadratic complexity w.r.t the point number, meaning that the complexity is $O(N^2)$ with N the point count. Some works proposed patch-based attention (Berg *et al.*, 2022; C. Zhang *et al.*, 2022), which partially alleviates this by reducing complexity to

$O(NM)$ with M the total number of patches. In contrast, our complexity is $O(NC)$, where C is a constant describing the number of classes. Importantly, C remains constant regardless of dataset scale and satisfies $C \ll M \ll N$. Last, equipped with the PE module, our network becomes single-stage and end-to-end trainable, without requiring any data preprocessing. In Section 4.4.6, we provide a quantitative analysis of the complexity and computational efficiency of our method.

Relation to contrastive prototype learning. Contrastive learning is an effective tool for supervised learning and few-shot learning (Khosla *et al.*, 2020; M. Liu *et al.*, 2022). Given a set of prototypes, the core of contrastive learning is to employ a contrastive loss, forcing each point embedding to be more similar to its positive (more relevant) prototypes and less similar to its negative prototypes. Such a contrastive loss serves as a soft constraint and implicitly encourages intra-class compactness. On the contrary, we opt for a more direct approach. By expanding and projecting the prototypes to point embeddings according to their feature dependencies, we explicitly construct a more descriptive embedding space, where each point is closer to its highly related prototypes and further away from the less relevant prototypes. In this way, both the intra-class compactness and the between-class discrepancy are naturally enhanced. Our approach is straightforward and does not require additional network supervision.

4.4. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed PE module on 3D semantic segmentation tasks for both indoor and outdoor environments, adopting several well-established point cloud learning networks as the baseline methods.

4.4.1. EVALUATION SETUP

We conducted a comprehensive analysis of the proposed PE module by benchmarking it against several established point cloud learning networks as our baselines, such as PointNet++ (Qi, Yi, *et al.*, 2017), KP-Conv (Thomas *et al.*, 2019), PointTransformer (H. Zhao *et al.*, 2021), and PointMetaBase (H. Lin *et al.*, 2023). Our PE module is designed with flexibility in mind, allowing seamless integration into existing architectures. Theoretically, it can be applied after each feature transformation layer of the network. In our experiments, we incorporated PE between the last decoding layer and the head layer.

We assessed the effectiveness of our PE module with two distinct tasks: indoor scene semantic segmentation and outdoor scene semantic segmentation. For the indoor segmentation task, we used the widely adopted S3DIS dataset (Armeni *et al.*, 2016). For the outdoor segmentation task, we utilized the more recent Toronto3D dataset (W. Tan *et al.*, 2020). The performance evaluation was conducted using standard metrics, including OA, mIoU, and per-category IoU scores. A detailed explanation of these metrics is provided in the previous Section 3.4.3.

	Settings	S3DIS				Toronto3D
		PointNet++	KP-Conv	PointTransformer	PointMetaBase	KP-Conv
Data	voxelization size	-	4cm			8cm
	batch sampling	block	sphere	whole scene	random cropping	sphere
	block/sphere radius	1.0m	1.2m	-		3.0m
	points per batch	4096	varying	max 80000	24000	varying
Training	optimizer	Adam	SGD		AdamW	SGD
	base learning rate	0.001	0.01	0.5	0.01	0.01
	scheduler	step decay		multi-step decay	cosine decay	step decay
	weight decay	0.0001	0.001	0.0001		0.001
	batch size	16	6	16	8	4
	steps per epoch	2973	500	30	765	500
	epochs	32	500	100		400

Table 4.1: Details of data preprocessing, sampling, and network training.

4.4.2. IMPLEMENTATION DETAILS

To ensure a fair comparison, we trained each baseline network and its PE-enhanced counterpart under the same experimental settings. Table 4.1 outlines the data preprocessing, point sampling strategies, and network training configurations employed for all baseline models in our experiments.

PointNet++ and PointMetabBase randomly sample a fixed number of points (e.g., 2048) for network training. Given the large-scale nature of the S3DIS and Toronto3D datasets, which contain dense point clouds that cannot be directly processed by the networks, most approaches perform voxelization with a dataset-specified grid size prior to training. Additionally, different networks adopt different strategies to sample points into batches. PointNet++, KP-Conv, and PointMetaBase sample partial point clouds using random 3D blocks, random 3D spheres, and random cropping of the input points, respectively. On the contrary, PointTransformer takes the entire scene as input to batches. We followed the same data preprocessing and sampling strategies as the corresponding baseline networks.

For all experiments, the coefficient for prototype update in Equation 4.1 was empirically set to $\delta = 0.8$. Training settings, including optimizer choice, learning rate schedule, weight decay, and number of epochs, were maintained consistently with those used by the original baseline methods.

4.4.3. RESULTS OF INDOOR SCENES

For the indoor scene semantic segmentation task, we used the challenging S3DIS dataset (Armeni *et al.*, 2016), which contains six building areas, 217 scenes, and 13 semantic categories (e.g., window, wall, table) in total. For this task, we adopted the four prominent networks as our baselines: PointNet++ (Qi, Yi, *et al.*, 2017), KP-Conv (Thomas *et al.*, 2019), Point Transformer (H. Zhao *et al.*, 2021), and PointMetaBase (H. Lin *et al.*, 2023). Among these baselines, PointNet++ and PointMetabase are MLP-based networks, KP-Conv employs 3D convolution, and Point Transformer utilizes a transformer-based framework. As detailed in Table 4.1, we followed the same data sampling, augmentation, and optimization schemes (e.g., optimizer, learning rate schedule, weight decay, and supervision loss) as the corresponding baseline networks for a fair evaluation.

Table 4.2 reports the performance comparison between the baselines and our PE-enhanced networks. Note that the official PointNet++ paper (Qi, Yi, *et al.*, 2017) did not report the scores in S3DIS. Hence, we used the results reported in its Pytorch implementation (X. Yan, 2019). In addition, PointMetabase (H. Lin *et al.*, 2023) reported the average scores over three training runs using the three seeds 2425, 4333, and 1111. To ensure a fair comparison, we adopted the same seed configuration and reported our averaged scores on three training runs.

From these quantitative results, we can see that our approach consistently surpasses all baselines, achieving mIoU gains of 2.4%, 0.9%, 0.1%, and 0.5% for

Method	OA(%)	**mIoU**(%)	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet (Qi, Su, *et al.*, 2017)	-	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
PointCNN (Y. Li *et al.*, 2018)	85.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph (Landrieu & Simonovsky, 2018)	86.4	58.0	89.4	96.9	78.1	0.0	42.8	48.9	61.6	75.4	84.7	52.6	69.8	2.1	52.2
MinkowskiNet (Choy *et al.*, 2019)	-	65.4	91.8	98.7	86.2	0.0	34.1	48.9	62.4	81.6	89.8	47.2	74.9	74.4	58.6
PatchFormer (C. Zhang *et al.*, 2022)	-	68.1	-	-	-	-	-	-	-	-	-	-	-	-	-
CBL (Tang *et al.*, 2022)	90.6	69.4	93.9	98.4	84.2	0.0	37.0	57.7	71.9	91.7	81.8	77.8	75.6	69.1	62.9
StratifiedFormer (Lai *et al.*, 2022)	91.5	72.0	96.2	98.7	85.6	0.0	46.1	60.0	76.8	92.6	84.5	77.8	75.2	78.1	64.0
PointNet++ (X. Yan, 2019)	83.0	53.5	89.4	97.7	**75.4**	0.0	1.8	58.3	**19.5**	69.2	79.0	46.2	59.1	58.7	41.6
+ PE	**83.6**	**55.9**	**90.6**	97.7	74.0	0.0	**10.5**	**58.5**	8.7	**72.8**	**83.4**	**55.1**	**66.5**	**61.3**	**47.3**
KP-Conv rigid (Thomas *et al.*, 2019)	-	65.4	92.6	97.3	81.4	0.0	16.5	54.5	**69.5**	80.2	90.1	66.4	74.6	63.7	58.1
+ PE	-	**66.3**	**93.7**	**98.6**	**82.4**	0.0	**21.2**	**58.3**	66.5	**82.0**	**91.3**	**67.0**	**74.9**	**67.7**	**58.6**
PointTransformer (H. Zhao *et al.*, 2021)	**90.8**	70.4	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	**74.3**	**80.2**	76.0	**59.3**
+ PE	90.6	**70.5**	**94.2**	**98.6**	**86.4**	0.0	**38.1**	**63.8**	**77.2**	**90.8**	**83.0**	72.2	76.7	**77.8**	57.7
PointMetaBase-L (H. Lin *et al.*, 2023)	90.5	69.5	94.8	98.3	84.1	0.0	28.5	**59.6**	75.7	81.7	**91.3**	74.4	**77.1**	76.2	**61.8**
+ PE	90.5	**70.0**	**94.9**	98.3	**84.3**	0.0	**33.9**	58.0	**77.0**	**82.8**	91.0	**75.5**	76.2	**76.9**	61.6

Table 4.2: Semantic segmentation results achieved on the S3DIS dataset (Armeni *et al.*, 2016), accessed in December 2023. OA, mIoU, and per-category IoU scores are reported. Note that PointMetaBase (H. Lin *et al.*, 2023) reported the average scores over three training runs using the three seeds 2425, 4333, and 1111. We used the same seed setting and reported the average scores on three training runs for a fair comparison. The numbers in **bold** highlight the better performance in the comparison between the baseline and PE-enhanced network. The underlined numbers denote the best performance.

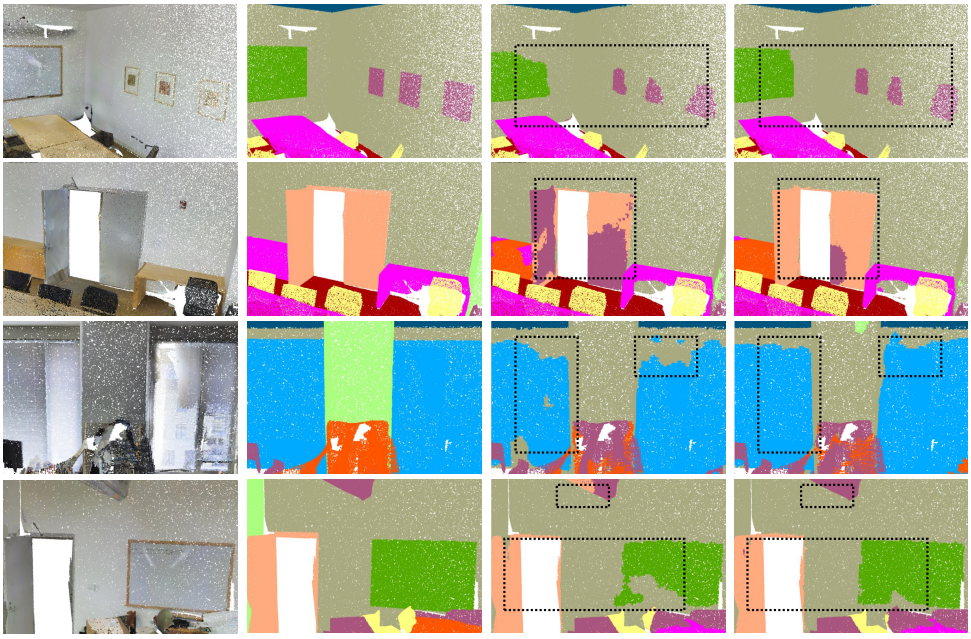


Figure 4.4: Qualitative results on the S3DIS dataset (Armeni *et al.*, 2016) test set Area 5, using KP-Conv (Thomas *et al.*, 2019) as the baseline. GT denotes Ground Truth. Our approach demonstrates cleaner segmentation results with higher objectiveness and more complete semantic boundaries, which are highlighted by the black-dotted boxes.

PointNet++, KP-Conv, Point Transformer, and PointMetaBase, respectively. These results indicate that our PE mechanism is effective across a diverse set of backbone architectures with varying feature learning paradigms. Notably, we observed a performance improvement in most semantic categories. For instance, 9 categories report a higher IoU score with PointNet++, 11 with KP-Conv, 9 with Point Transformer, and 7 with PointMetaBase. Such performance improvements suggest that the proposed PE module enhances the general feature representation capability of 3D point clouds, rather than being tailored to specific semantic classes.

In Figure 4.4, Figure 4.5, and Figure 4.6, we present qualitative results achieved on the S3DIS dataset using KP-Conv (Thomas *et al.*, 2019), PointMetaBase (H. Lin *et al.*, 2023) and Point Transformer (H. Zhao *et al.*, 2021) as baseline architectures, respectively. Similar to Chapter 3, we use the following color schemes to render the scenes in the three figures: for *ceiling*, for *floor*, for *wall*, for *beam*, for *column*, for *window*, for *door*, for *table*, for *chair*, for *sofa*, for *bookcase*, for *board*, and for *clutter*.

In all of these visual comparisons, the integration of our proposed PE module

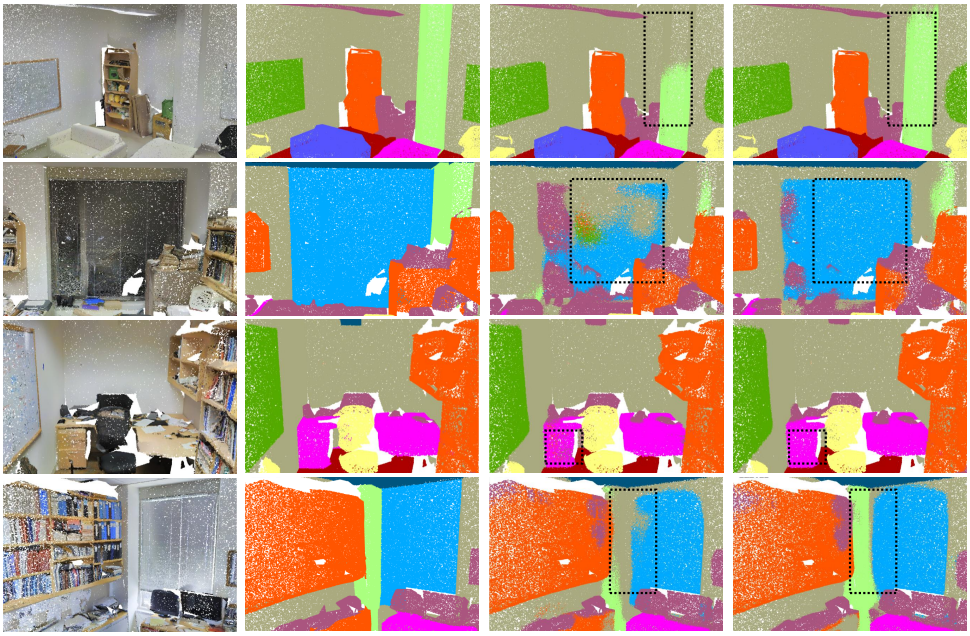


Figure 4.5: Qualitative results on the S3DIS dataset (Armeni *et al.*, 2016) test set Area 5, using PointMetaBase (H. Lin *et al.*, 2023) as the baseline. Our improvements are highlighted with black-dotted boxes.

leads to notable enhancements in segmentation quality. In particular, our PE module demonstrates significant improvements in the identification and segmentation of various object categories, such as *window* and *table*. It is evident that the PE-enhanced network effectively reduces semantic segmentation errors by mitigating segmentation inconsistencies and capturing objects with more complete geometrical shapes. This improvement can be attributed to the effectiveness of the *Expansion* operator, which performs global prototype expansion and projection to refine pointwise embeddings through point-to-prototype interaction.

Furthermore, the PE module demonstrates the capability to detect minority classes such as *Column* and *Door*, which constitute only a small fraction of the training dataset. The main reason is that the global prototype expansion naturally encourages intra-class compactness, thereby facilitating more discriminative feature learning without the need for additional supervisory loss terms. However, as discussed in Section 4.4.7, the integration of PE may occasionally introduce segmentation noise in local regions. This is also indicated in Figure 4.6, row 3, where PE introduces minor segmentation noise in the category of *bookcase*. Overall, both quantitative experiments and qualitative visualizations indicate that our proposed PE module can effectively enhance segmentation performance across diverse backbone architectures and feature learning paradigms.

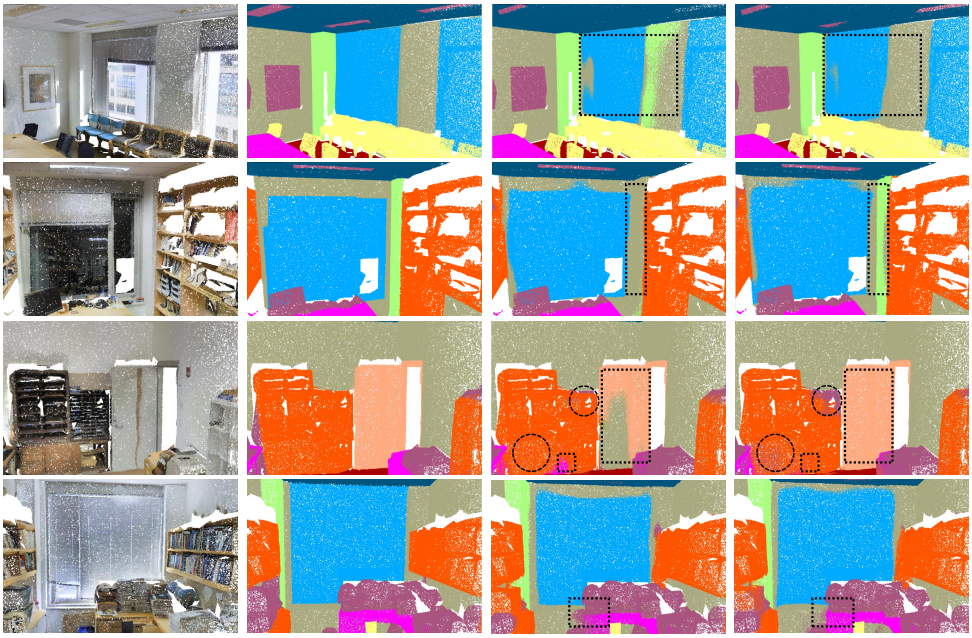


Figure 4.6: Qualitative results on the S3DIS dataset (Armeni *et al.*, 2016) test set Area 5, using PointTransformer (H. Zhao *et al.*, 2021) as the baseline. Our improvements are highlighted with black-dotted boxes. Meanwhile, the minor segmentation noise in some local regions is marked by black-dotted circles.

4.4.4. RESULTS OF OUTDOOR SCENES

To further validate the generalizability of the proposed PE module across diverse real-world environments, we evaluated its performance on the Toronto3D dataset (W. Tan *et al.*, 2020), a large-scale urban point cloud dataset captured via mobile laser scanning systems along Avenue Road in Toronto, Canada. Toronto3D comprises 78.3 million points spanning approximately 1 km of urban streets, and is divided into four distinct areas annotated with eight semantic classes such as road, natural, pole, and car. Following the standard evaluation guideline of the Toronto3D benchmark, we used areas L001, L003, and L004 for training, and L002 for testing. Our evaluation employed KP-Conv (Thomas *et al.*, 2019) as our baseline, training with the additional RGB values as the input. Due to the performance variations of KP-Conv on this dataset, we reported the averaged scores over three training runs for both the baseline and our PE-enhanced model.

As presented in Table 4.3, the integration of the proposed PE module exhibits a notable 1.0% improvement in mIoU, achieving performance on par with other state-of-the-art methods. Five categories demonstrate superior IoU scores compared

Method	OA(%)	**mIoU**(%)	road	road mark	natural	building	utility line	pole	car	fence
PointNet++ (Qi, Yi, *et al.*, 2017)	84.9	41.8	89.3	0.0	69.0	54.1	43.7	23.3	52.0	3.0
DGCNN (Y. Wang *et al.*, 2019)	94.2	61.8	93.9	0.0	91.3	80.4	62.4	62.3	88.3	15.8
MS-PCNN (L. Ma *et al.*, 2019)	90.0	65.9	93.8	3.8	93.5	82.6	67.8	72.0	91.1	22.5
MS-TGNet (W. Tan *et al.*, 2020)	95.7	70.5	94.4	17.2	95.7	88.8	76.0	74.0	94.2	23.6
RandLa-Net* (Q. Hu *et al.*, 2020)	94.4	81.8	96.7	64.2	96.9	94.2	88.1	77.8	93.4	42.9
MappingConvSeg* (K. Yan *et al.*, 2021)	94.7	82.9	97.2	67.9	97.6	93.8	86.9	82.1	93.7	44.1
KP-Conv* (Thomas *et al.*, 2019)	97.5	79.1	97.3	59.2	96.7	93.1	87.0	**83.0**	**95.5**	20.8
+ PE	**97.7**	**80.1**	**97.5**	**65.7**	96.7	**93.3**	**87.2**	82.4	95.2	**22.9**

Table 4.3: Semantic segmentation results achieved on the Toronto3D dataset (W. Tan *et al.*, 2020), evaluated on L002, accessed in December 2023. OA, mIoU, and per-category IoU scores are reported. Methods denoted with * also use RGB values as network input. Scores of the six competing networks are obtained from the benchmark (W. Tan *et al.*, 2020). Due to performance variations, scores of KP-Conv (Thomas *et al.*, 2019) and our PE-enhanced network are averaged over three training runs. The numbers in **bold** highlight the better performance in the comparison between the baseline and PE-enhanced network. The underlined numbers denote the best performance.

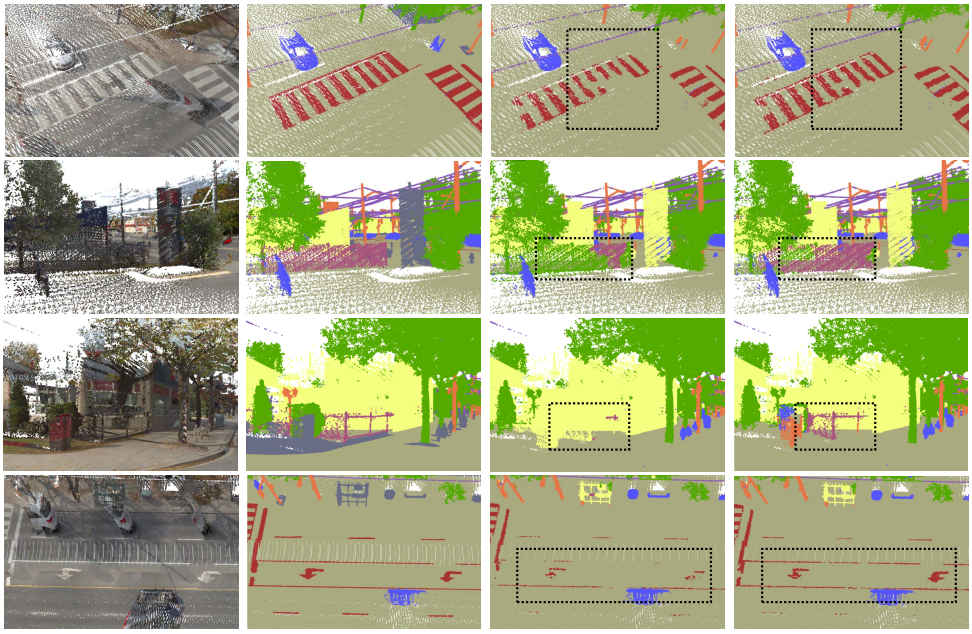


Figure 4.7: Qualitative results on the Toronto3D dataset (W. Tan *et al.*, 2020) test set L002, using KP-Conv (Thomas *et al.*, 2019) as the baseline. Our improvements are highlighted with black-dotted boxes. We use the following colors to render distinct urban objects: for *road*, for *road mark*, for *natural objects*, for *building*, for *utility line*, for *pole*, for *car*, and for *fence*.

to the baseline, underscoring the effectiveness of PE. Particularly noteworthy are the significant enhancements observed in minor classes. For instance, the categories of *road mark*, *utility line*, and *fence* collectively account for only 2.3%, 0.8%, and 0.5% of the whole dataset. The scarcity of training data for these minor classes poses challenges for existing methods in accurately detecting them. Despite this, the PE module achieved substantial improvements in IoU scores for these categories, with gains of 6.5%, 0.2%, and 2.1%, respectively. These results demonstrate the potential of PE to alleviate the class imbalance issue and enhance feature learning for rare categories, without the need to design special loss terms to balance point count variations among different classes.

Figure 4.7 presents the segmentation results achieved on Toronto3D (W. Tan *et al.*, 2020), adopting KP-Conv (Thomas *et al.*, 2019) as the baseline. The proposed PE module enhances segmentation performance by better preserving the geometric completeness of object shapes. Specifically, we have observed significant improvements in the segmentation of categories such as *fence* and *road mark*, highlighting the module's capacity in addressing class imbalance without the need

for special supervisory loss terms. This indicates that our PE module effectively enhances the segmentation of minor classes, showing its ability to alleviate the class imbalance issue prevalent in 3D semantic segmentation tasks.

4.4.5. ABLATION STUDIES

The design of our PE module diverges from the conventional *Attention* mechanism (Vaswani *et al.*, 2017) in two key aspects: First, it utilizes the inner product of raw features, rather than transformed representations such as keys and queries, to model point-to-prototype dependencies. Second, instead of incorporating a residual link (He *et al.*, 2016) for feature recalibration, our approach incorporates a gating mechanism. In this section, to validate the contribution of each design component, we conducted a series of ablation studies. These experiments were performed on the S3DIS dataset (Armeni *et al.*, 2016), using PointMetaBase (H. Lin *et al.*, 2023) as the baseline network. For consistency and fair comparison, all experiments were conducted with a fixed random seed set to 2425. Table 4.4 summarizes the results of our ablation studies.

Method	mAcc(%)	mIoU(%)
Baseline network	75.9	69.6
Using embedded feature inner product	75.2	69.0
Using ReLU + residual link	74.7	68.6
Using ReLU + Hadamard product	76.0	69.7
Using Sigmoid + Hadamard product (PE)	**77.3**	**70.5**

Table 4.4: mAcc and mIoU scores of the ablated networks. The experiments were carried out using the seed 2425.

Modeling point-to-prototype dependencies. As shown in Table 4.4, replacing the raw feature inner product with the transformed feature inner product (e.g., inner product between keys and queries) leads to a performance decline, with mAcc and mIoU decreasing by 0.7% and 0.6%, respectively, relative to the baseline. In contrast, our proposed use of the raw feature inner product yields performance gains, improving mAcc by 1.4% and mIoU by 0.9% over the baseline.

Feature recalibration. We investigate three alternative strategies for feature recalibration in Equation 4.4: (i) Using ReLU with a residual link. (ii) Using ReLU with Hadamard product. (iii) Using Sigmoid with Hadamard product. Experimental results indicate that the combination of Sigmoid activation and Hadamard product, which corresponds to our proposed method, yields the best performance. Specifically, it outperforms variants (i) and (ii) by 1.9% and 0.8% in mIoU, demonstrating its superior effectiveness among the three configurations.

4.4.6. COMPLEXITY AND EFFICIENCY

To evaluate the model complexity and computational efficiency of our PE-enhanced networks, we conducted a comparative analysis against the baseline networks using two primary metrics: the number of parameters and the inference speed, measured in terms of Throughput (TP). All experiments were performed under identical batch configurations across networks on an NVIDIA RTX A6000 GPU to ensure a fair comparison. For the S3DIS benchmark, we set the batch size to 16 and the number of points per batch to 15,000. The detailed results are presented in Table 4.5.

	Methods	Params (M)		TP (ins./sec)	
		Baseline	+ PE	Baseline	+ PE
S3DIS	PointNet++	0.97	0.97	37.48	36.65
	KP-Conv Rigid	25.59	25.61	-	-
	PointTransformer	7.77	7.77	59.86	58.91
	PointMetaBase-L	2.74	2.74	132.13	144.32
Toronto3D	KP-Conv Rigid	24.38	24.40	-	-

Table 4.5: Comparison of the number of network parameters and TP between baselines and our PE-enhanced networks. All measurements are conducted with an NVIDIA RTX A6000 GPU.

Our PE module results in a minimal increase in model complexity relative to the baseline networks, introducing only 0.005M and 0.018M additional parameters on average for the S3DIS and Toronto3D datasets. For the inference speed, our PE-enhanced networks achieve a similar Throughput as the baseline networks, with only a minor trade-off. This marginal overhead can be attributed to two main design choices: First, we use a non-parametric approach for prototype construction and updating, which avoids additional learnable parameters. Second, we employ a simple and lightweight *Expansion* step, implemented using only a one-layer MLP as the projection function $h(\cdot)$, as defined in Equation 4.3 and illustrated in Figure 4.3. All remaining operations within the PE module are parameter-free, further contributing to its efficiency.

Considering the performance improvements achieved by the proposed PE module on both tasks (see Tables 4.2 and 4.3), along with its minimal impact on inference time, we conclude that the PE module consistently enhances baseline models in an efficient and computationally lightweight manner.

4.4.7. LIMITATIONS

Our proposed *Prototype Expansion* module facilitates efficient global feature analysis by leveraging class prototypes and explicitly expanding them into point-level embeddings based on their relationships within the feature space. This

approach effectively enhances intra-class feature compactness, leading to improved performance across a range of baseline models. However, despite these advantages, PE has several limitations that require further investigation.

First, the current implementation of the PE module employs a single prototype to represent each semantic class. While computationally efficient, this simplification overlooks intra-class variability, potentially ignoring the diverse distribution of point embeddings within the same category. Second, the PE design does not explicitly model spatial relationships between individual points and class prototypes. In particular, positional encoding is not incorporated in the *Expansion* step, despite the well-established importance of spatial information in 3D vision tasks. Finally, although the explicit injection of global semantic information into individual point-level embeddings enhances overall structural understanding, it may have difficulty in handling local regional details. This can result in misclassifications or noisy predictions in spatially complex regions, as illustrated in the third scene of Figure 4.6.

4.5. CONCLUSIONS

While the previous Chapter 3 focuses on local semantic refinement near object boundaries, in this chapter, we investigate the task of 3D semantic segmentation from a global perspective. We have presented *Prototype Expansion*, a novel network module designed to enhance the representational capacity of individual point features through point-to-prototype interactions. Unlike conventional approaches that often rely on local network operators with constrained receptive fields, our PE block leverages global prototype descriptors to recalibrate individual point embeddings. This mechanism effectively transforms the PE block into a global feature operator, thereby contributing to the improved intra-class compactness.

Comprehensive experiments on both indoor scene segmentation and outdoor scene segmentation tasks, across multiple baseline architectures, have demonstrated the effectiveness and generalizability of the proposed PE module. These results underscore the utility of incorporating global context, positioning the PE module as a meaningful contribution to 3D point cloud processing tasks.

5

STRUCTURE-AWARE TREE INSTANCE SEGMENTATION

While Chapters 3 and 4 focus on 3D semantic segmentation, this chapter advances the discussion toward fine-grained instance-level segmentation of urban environments, with a particular emphasis on trees. Trees are vital components of urban planning, forestry, and ecological systems. However, unlike buildings that can often demonstrate regular outlines and can be delineated using existing footprint data, trees pose significant segmentation challenges due to their complex nature and geometric variabilities. In this chapter, we introduce a novel structure-aware method for robust instance segmentation of urban trees directly from 3D point clouds. The proposed approach employs a multi-task learning framework that simultaneously performs (i) semantic segmentation to classify a point as crown, stem, or other; (ii) heatmap prediction to assign a heat value to each point based on 2D Gaussian kernels centered at tree stem locations; (iii) offset prediction to estimate point-wise offset vectors pointing to the instance centroid. These outputs are integrated to accurately localize tree stems. Then, we adopt a graph-based shortest path algorithm to group individual tree points based on the localized stems. Extensive experiments on two public forestry datasets, TreeML and ForInstance, demonstrate that our approach consistently outperforms state-of-the-art methods, reducing significant under-segmentation or over-segmentation errors. Our segmentation framework has broad applicability across various downstream domains, including urban landscape design, carbon cycle modeling, and environmental simulation.

This chapter is based on the paper: Shenglan Du, Jantien Stoter, Julian F.P. Kooij, Liangliang Nan. SATree: Structure-aware Tree Instance Segmentation from 3D LiDAR Point Clouds. Urban Forestry & Urban Greening 120 (2026): 129414. DOI: 10.1016/j.ufug.2026.129414.

5.1. INTRODUCTION

3D instance segmentation, a more fine-grained task than 3D semantic segmentation, seeks to partition urban scenes into distinct object instances. This chapter focuses on 3D instance segmentation, with a specific emphasis on large-scale trees. Trees play a critical role in urban ecosystems, providing significant ecological and economic benefits by contributing to photosynthetic activity, maintaining carbon balance, and regulating the temperature (Hyyppä *et al.*, 2012). Unlike buildings, which typically exhibit regular outlines and simple geometries, trees present greater challenges due to their inherently complex and variable morphology. Accurate quantification and assessment of trees is a fundamental task for many applications such as urban planning, forestry management, and environmental simulations. For example, estimating forest biomass and volume at the individual tree level is crucial for accurate carbon storage assessments, which further helps to develop climate change mitigation strategies (Shrestha *et al.*, 2018); Modeling trees within urban green spaces is also important to landscape architects and urban designers, contributing to the planning and sustainability of modern cities. All the aforementioned applications require precise tree inventory at the instance level.

Traditional inventories of trees and vegetation heavily rely on field surveys, which can be labor-intensive, time-consuming, and costly (Hyyppä *et al.*, 2001). With recent advances in remote sensing technology, 2D satellite imagery and 3D LiDAR data have been used to efficiently characterize forest structure in large areas (Dassot *et al.*, 2011). Specifically, a wide range of image-based AI approaches have been proposed to facilitate vegetation segmentation (Arief *et al.*, 2018), identify tree species (Hakula *et al.*, 2023), delineate individual tree crowns (Weinstein *et al.*, 2020; Yun *et al.*, 2021), and analyze tree structures (Reche-Martinez *et al.*, 2004). However, images may suffer from low resolution, weather sensitivity, and occlusion issues that hinder accurate data acquisition (P. Wang *et al.*, 2023). Furthermore, imagery naturally does not capture tree structure details, thus failing to achieve fine-grained tree instance segmentation, particularly in dense forest areas with complex tree shapes. Compared to images, LiDAR point clouds directly capture object surfaces with accurate 3D measurements and rich geometrical details. Given their high spatial resolution and accuracy, LiDAR data have been widely adopted for individual tree segmentation, which further enables researchers to derive key botanical structural parameters such as tree height (Olofsson *et al.*, 2014), Diameter at Breast Height (Sun, Jin, *et al.*, 2022), as well as tree volume and biomass (Fan *et al.*, 2020).

Individual tree segmentation from LiDAR point clouds can be categorized into heuristic- and learning-based approaches. Heuristic-based methods often assume that tree tops are local maxima and thus can be detected by watershed algorithms (Q. Chen *et al.*, 2006). Another common assumption is that tree crown points form dense clusters of points in the 3D Euclidean space. Therefore, they can be segmented through various clustering techniques, including mean-shift clustering (Malladi *et al.*, 2024), density-based clustering (Hakula *et al.*, 2023; J. Wang *et al.*, 2018), hierarchical clustering (Lee *et al.*, 2010), and graph shortest path algorithm (Livny *et al.*, 2010;

Tao *et al.*, 2015). However, these methods highly demand domain-specific heuristic knowledge as priors. On the other hand, driven by the success of deep learning in computer vision and point cloud analysis, several learning-based methods have been introduced, aiming to address the limitations of conventional approaches. Early methods typically convert point clouds into discretized models such as CHMs or DSMs and apply image processing techniques, e.g., CNNs, to achieve individual tree segmentation (Chang *et al.*, 2022; Hamraz *et al.*, 2019; J. Wang *et al.*, 2019). The 2D segmentation results are then projected back to the original 3D space to obtain 3D tree instances. Such data transformation between 2D and 3D typically introduces information loss. To address this issue, more recent approaches (Henrich *et al.*, 2024; T. Jiang, Liu, *et al.*, 2023; T. Jiang, Wang, *et al.*, 2023; H. Luo *et al.*, 2021; Z. Luo *et al.*, 2021; P. Wang *et al.*, 2023) directly process point clouds and perform per-point instance predictions in the 3D space. Among them, P. Wang *et al.* (2023) designs a two-branch network that fuses the features of the semantic and instance branches for tree instance segmentation. Henrich *et al.* (2024), T. Jiang, Liu, *et al.* (2023), and H. Luo *et al.* (2021) perform joint semantic segmentation and offset prediction (i.e., a directional vector pointing to the tree instance centroid), followed by clustering the points into individual trees. Notably, TreeLearn (Henrich *et al.*, 2024) utilizes the state-of-the-art 3D instance segmentation network SoftGroup (Vu *et al.*, 2022) as its backbone. Similarly, T. Jiang, Wang, *et al.* (2023) jointly learns point semantics and offsets but also integrates tree centroid prediction as the second stage to enhance individual tree segmentation.

Despite their impressive performances, these methods struggle to accurately identify individual trees in complex urban forest environments, where trees exhibit significant overlap or occlusion. This often leads to under-segmentation or over-segmentation errors. To address these challenges, we propose SATree, a Structure-Aware Tree instance segmentation approach targeting challenging urban forestry areas. By detecting critical tree structures such as stems and crowns, we can achieve precise tree instance segmentation that is robust against tree shape complexities and varying sizes.

The idea of localizing tree stems to find individual trees has been recently explored by Ning *et al.* (2023), Pu *et al.* (2023), and J. Wang *et al.* (2019). However, they regard stem detection as a separate task, which often requires additional network learning modules, thus adding extra data preprocessing and computational complexity. In contrast, we introduce a simple and unified point cloud learning framework that identifies both tree crowns and stems in a parallel manner. Figure 5.1 gives an approximate overview of our workflow.

We use one network to simultaneously perform three tasks: (i) Semantic segmentation. Each point is classified as *crown*, *stem*, or *other*; (ii) Heatmap prediction. A heat value is predicted for each point based on 2D Gaussian kernels centered at tree stem locations. The heat value increases as the point approaches the tree stem or main branches, to support more accurate stem localization; (iii) Offset prediction. A 3D offset vector is predicted for each point, directing it toward its corresponding tree instance centroid. This unified framework enables high-fidelity

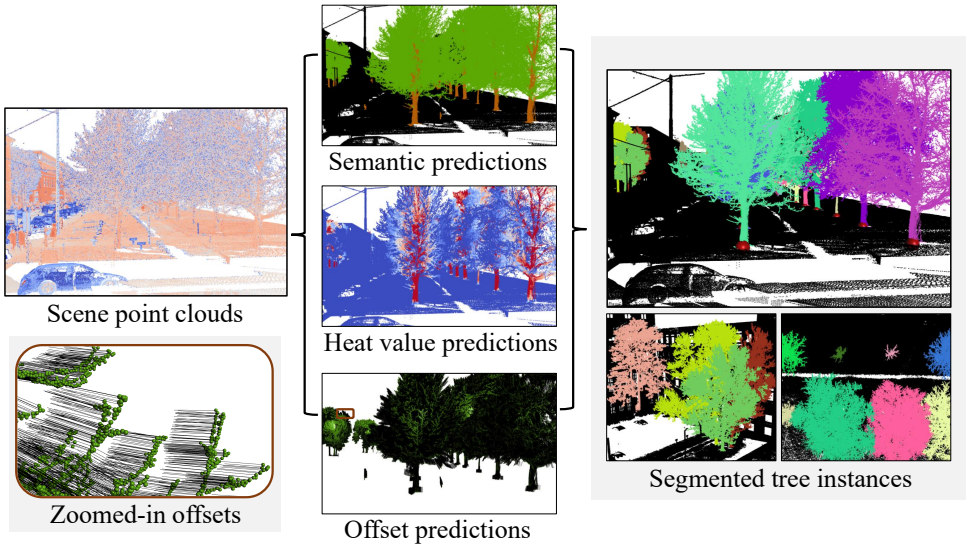


Figure 5.1: Our workflow with three intermediate outputs and final tree instances. We simultaneously perform three tasks: (i) semantic segmentation to classify if a point is *crown* (green), *stem* (brown), or *other* (black); (ii) heatmap prediction to assign a heat value to each point, ranging from 0 (blue) to 1 (red). High values indicate tree stems or main branches; (iii) offset prediction to predict an offset vector pointing to the corresponding tree instance centroid. At the bottom left, we visualize the zoomed-in predicted offsets. Tree points are visualized in green, and offset vectors are visualized with black lines. By combining the three task predictions, we achieve accurate tree instance segmentation. At the bottom right, we can effectively segment trees of various sizes in challenging environments, demonstrating the robustness of our proposed approach in complex urban forestry scenes.

localization of tree stems without any preprocessing or separate network learning processes. Once tree stems are identified, we use a graph-based shortest path approach to isolate individual trees, which considers the proximity of points in the Euclidean space and their offset orientation towards the instance centroids. This strategy helps to precisely delineate tree instance boundaries.

Our work provides two major contributions:

- A multi-task learning framework that jointly segments crowns and stems from the urban scenes. In addition to that, we also involve heatmap prediction and offset prediction to guide 3D tree instance segmentation.
- A graph-based shortest path method that incorporates the learned offset embeddings for precise delineation of tree boundaries in complex areas.

5.2. RELATED WORK

While Section 2.3 has reviewed a large number of 3D instance segmentation methods targeting general urban objects, in this section, we review the methods specifically tailored for individual tree detection and tree instance segmentation, ranging from conventional heuristic-based approaches to recent learning-based approaches.

5.2.1. HEURISTIC-BASED APPROACHES

Automatic segmentation of individual trees is challenging due to the irregularity of tree shapes and the inherent complexity of forest ecosystems. Early approaches primarily focused on identifying and extracting treetops from 2D aerial imagery (Dralle & Rudemo, 1996; Wulder *et al.*, 2000), which was later extended to 3D LiDAR data (Q. Chen *et al.*, 2006). Following treetop detection, classical image segmentation techniques, such as watershed segmentation (Beucher, 1979), can be applied to recognize individual trees. To achieve more accurate segmentation of tree morphological shapes, Yun *et al.* (2021) introduced a dual Gaussian filter in combination with an anisotropic water expansion algorithm for crown boundary segmentation. However, such methods simply detect treetops as local maxima from images or CHMs, which can result in large commission errors (Q. Chen *et al.*, 2006).

Another line of approaches adopts grouping-based strategies to segment individual trees from point clouds, based on the observation that tree canopy points naturally form dense clusters in the 3D space. The clustering algorithms used for grouping the points are k-means (Gupta *et al.*, 2010), hierarchical clustering (Lee *et al.*, 2010), and mean-shift (Malladi *et al.*, 2024). Studies of Ayrey *et al.* (2017), Hakula *et al.* (2023), and J. Wang *et al.* (2018) propose a layer-wise stacking strategy to mitigate clustering errors in dense forest areas. These methods slice the forest canopy into layers, cluster points per layer, and aggregate the layer-wise clusters into tree instances. Several works also leverage graph structure for tree instance segmentation. Works of Livny *et al.* (2010) and Tao *et al.* (2015) applied the graph shortest path algorithm to group individual trees. Heinzel and Huber (2018) constructed a similarity graph over the points and segmented the trees using a Markov Random Field framework. D. Wang *et al.* (2021) developed a hybrid approach combining the Delaunay graph and kNN graph, where each node repeatedly walks to its lowest neighbor to locate its tree source.

A major limitation of heuristic-based approaches is that they heavily rely on domain-specific knowledge as priors, making them difficult to generalize to a broad range of forestry scenes. Finding the optimal solution is often non-trivial for a specific urban scene or tree type.

5.2.2. DEEP LEARNING-BASED APPROACHES

Enabled by the broad applications of deep learning in computer vision and point cloud analysis, numerous learning-based approaches have been developed to address

tree instance segmentation.

A number of studies have directly applied image-based object detection networks to identify 2D urban tree instances from captured RGB imagery. Among them, DeepForest (Weinstein *et al.*, 2019, 2020) adopts the RetinaNet (T.-Y. Lin *et al.*, 2017) detector to produce tree bounding box predictions from aerial images. Follow-up studies have adapted R-CNNs, e.g., Faster R-CNN and C-Mask R-CNN, for individual tree detection and counting (Osco *et al.*, 2020; Sun, Li, *et al.*, 2022). Ammar *et al.* (2021) compared several CNN-based object detection approaches for palm tree counting in large farm areas, concluding that Yolov4 (Bochkovskiy *et al.*, 2020) and EfficientDet-D5 (M. Tan *et al.*, 2020) provide the best trade-off between accuracy and inference speed.

Besides, many research efforts have also been dedicated to 3D tree instance segmentation from LiDAR point clouds. Among them, discretization-based methods first discretize the tree points into grid-based models (e.g., DSMs and CHMs), then adopt a 2D object detection network to locate tree bounding boxes. Chang *et al.* (2022) projected tree points onto the ground plane and utilized Yolov3 (Redmon, 2018) for tree instance segmentation. Xi and Hopkinson (2021) transformed tree points into bird's-eye-view images and employed CenterNet (Duan *et al.*, 2019) for individual tree detection. Alternatively, point-based methods directly perform instance segmentation on 3D tree points, avoiding the potential information loss during data transformation. P. Wang *et al.* (2023) proposed a two-branch network that fuses features from the semantic branch and the instance branch to segment tree instances. X. Chen *et al.* (2021) used PointNet (Qi, Su, *et al.*, 2017) to classify tree points and obtained individual tree crown boundaries by analyzing height gradients. H. Luo *et al.* (2021) first performed semantic segmentation of trees, then incorporated an additional network to predict pointwise offset vectors directed towards object centroids, allowing points to aggregate into distinct tree instances. This strategy was later extended to the study of T. Jiang, Wang, *et al.* (2023), where tree centroids are mined from the learned offset embeddings to enhance the tree instance segmentation accuracy. Recently, Segmentanytree (Wielgosz *et al.*, 2024) and TreeLearn (Henrich *et al.*, 2024) adopted state-of-the-art 3D instance segmentation networks, e.g., PointGroup and SoftGroup, for individual tree segmentation. Additionally, several studies (Pu *et al.*, 2023; J. Wang *et al.*, 2019) focused on detecting tree stems to separate single trees, relying on transforming stem points to CHMs and detecting them using 2D image processing techniques.

Similar to previous studies, our proposed SATree performs joint semantic segmentation and offset prediction. Nevertheless, SATree explicitly detects main tree parts such as stems and crowns. Using these structures as priors, SATree can precisely localize, identify, and delineate individual trees even in challenging urban areas, maintaining its robustness against tree overlaps and varying tree sizes.

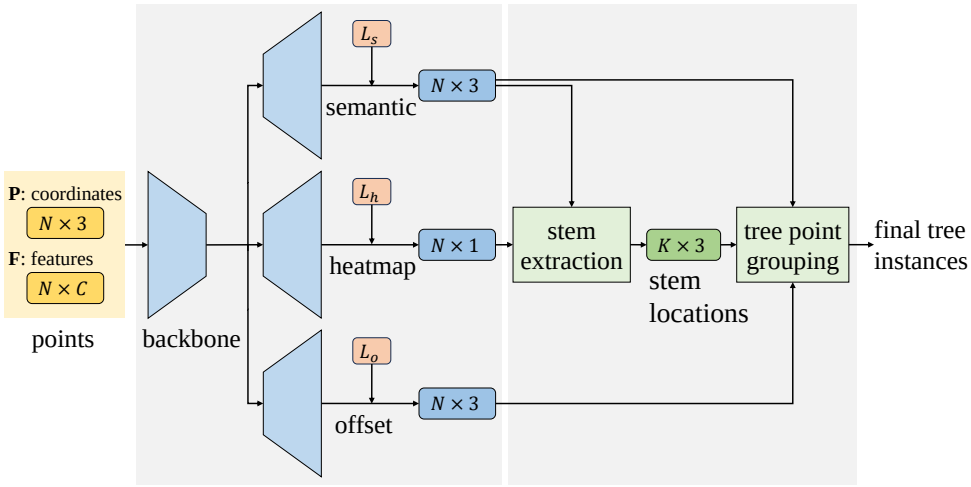


Figure 5.2: Framework of SATree. The input to our method is a point cloud containing coordinates and additional features. N is the number of input points. We use intensity as the input feature across all the experiments, i.e., $C = 1$. SATree has one shared point feature encoder and three decoders that jointly perform semantic segmentation, Gaussian heatmap prediction, and offset vector prediction. We use three distinct losses L_s, L_h, and L_o to supervise the corresponding tasks. Then, we combine the semantic outputs with the heatmap outputs to accurately localize tree stems. Lastly, the tree points are grouped based on the detected tree stem locations to generate the 3D tree instances.

5.3. METHOD

Our input is a point cloud $P \in R^{N\times(3+C)}$ captured from an urban scene containing both trees and non-tree objects, where N denotes the number of points and C denotes the input point feature dimension. Theoretically, any useful geometrical or spectral attributes can be used as input point features. In this study, we use the point intensity value as the feature for its simplicity and wide applicability, i.e., $C = 1$. Taking P, we use a point-based deep learning backbone to encode point features, followed by three decoding branches:

- a semantic segmentation branch for segmenting points into categories of *crown*, *stem*, and *other*;
- a Gaussian heatmap prediction branch for assigning a heat value to each point. We generate a 2D Gaussian-based heatmap on the x-y plane with peaks at the locations of individual tree stems. As detailed in Section 5.3.1, points near tree stems or main branches are predicted with high heat values, which helps to localize tree roots precisely;
- an offset prediction branch for predicting a pointwise offset vector directing

towards tree instance centroids.

The overall framework is illustrated in Figure 5.2. We can identify high-fidelity tree stem locations using the outputs from semantic segmentation and heatmap predictions. Subsequently, combining the outputs from all three branches, we group tree points based on the extracted stem locations to obtain the final tree instances.

5.3.1. NETWORK ARCHITECTURE

Our network has one shared feature encoder and three separate decoding branches. We use PointMetaBase (H. Lin *et al.*, 2023) as our backbone for the feature encoder to obtain high-level point features. In theory, any point-based learning networks (e.g., PointNet++ (Qi, Yi, *et al.*, 2017), KP-Conv (Thomas *et al.*, 2019), Point transformer (H. Zhao *et al.*, 2021), Stratified transformer (Lai *et al.*, 2022)) can be used. In this work, we choose PointMetaBase since it achieves a good trade-off between performance accuracy and computational efficiency. Following the shared feature encoder, the network consists of three decoding branches: the semantic segmentation branch, the heatmap prediction branch, and the offset prediction branch. Each branch performs a distinct task, which is detailed as follows:

SEMANTIC SEGMENTATION BRANCH

This branch outputs a semantic logit map $S \in R^{N \times K}$, where K is the number of semantic categories. In this study, the categories include *crown*, *stem*, and *other*, and thus $K = 3$. Tree crowns and stems are treated as distinct classes, while all the rest points in the scene (e.g., building, road, lamppost, pedestrian) are categorized as *other*, given our focus on tree objects. The high class imbalance poses challenges for standard supervision. For the urban environment, *other* (e.g., grounds, buildings, roads) accounts for the majority of the dataset while *stem* only accounts for a very small portion. Therefore, we use the weighted Cross Entropy loss to supervise this branch, where we assign a significantly higher weight to the class *stem* and a lower weight to *other*.

$$L_s = -\sum_{i=1}^{N} w_k \log p_i^k, \tag{5.1}$$

where N is the total number of points, k is the GT semantic label of the i^{th} point, p_i^k is the predicted probability of the i^{th} point belonging to its GT category that can be obtained from the network softmax layer, and w_k is the weight of the class k.

HEATMAP PREDICTION BRANCH

To better localize tree stems, we produce a 2D Gaussian-based heatmap on the x-y plane with peaks at the locations of individual tree stems. Then, we use this branch

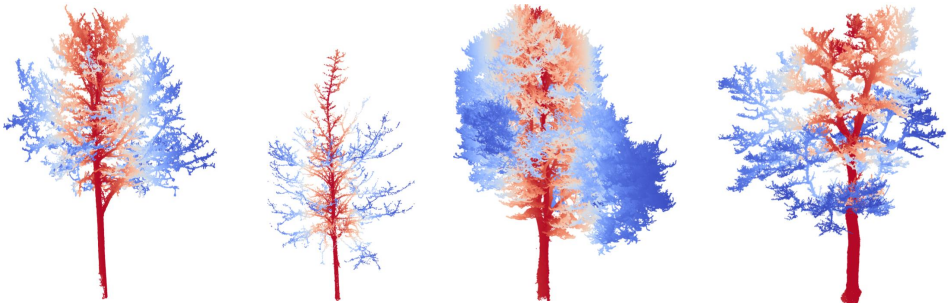


Figure 5.3: Visualization of GT Gaussian heatmaps of four individual tree instances with varying sizes and species. The Gaussian heat value ranges from 0 to 1, from blue to red. True stem points typically have higher heat values.

to predict a Gaussian heatmap $H \in R^{N\times 1}$. The GT heatmap $\hat{H}$ is created by retrieving each tree instance and placing a 2D Gaussian kernel centered at its respective tree stem location, which is defined as the geometrical center of the tree stem points. The GT heat value of the i^{th} point is computed as follows:

$$\hat{h_i} = e^{-\alpha d_i / r}, \tag{5.2}$$

where d_i is the 2D x-y distance from the i^{th} point to stem centroid of its corresponding tree instance. r is the Gaussian kernel radius defined as the maximum point-to-stem 2D distance we can obtain from the current tree instance. α is the hyperparameter that scales the Gaussian distribution. Figure 5.3 visualizes the GT heatmaps of several tree instances. All background points are assigned a heat value of 0. In such a way, we ensure that only points associated with tree stems or main structures are highlighted. The GT Gaussian heatmap only needs to be generated for the training dataset.

We use the MSE loss to supervise the heatmap branch:

$$L_{\text{h}} = \sum_{i=1}^{N} |h_i - \hat{h}_i|^2, \tag{5.3}$$

where h_i is the predicted heat value of the i^{th} point and $\hat{h}_i$ is the corresponding GT value.

OFFSET PREDICTION BRANCH

Following the standard practice in previous studies (Henrich *et al.*, 2024; T. Jiang, Liu, *et al.*, 2023; T. Jiang, Wang, *et al.*, 2023; H. Luo *et al.*, 2021; Wielgosz *et al.*, 2024), we use this branch to output an offset map $O \in R^{N\times 3}$, where for each point, we predict a 3D offset vector pointing to its tree instance centroid. The predicted offset vectors

are further used in the tree point grouping step to enhance the segmentation of tree instances. We use MSE loss to supervise this task, which is formulated as follows:

$$L_{\mathrm{o}} = \sum_{i=1}^{N} \|\mathbf{d_i} - \hat{\mathbf{d}}_\mathbf{i}\|_2^2, \tag{5.4}$$

where $\mathbf{d_i} \in R^3$ is the predicted offset direction vector. $\hat{\mathbf{d}}_\mathbf{i}$ is the GT offset vector of the i^{th} point. During training, the background (i.e., *other*) points are masked out, whereas only the tree points (i.e., *crown* and *stem*) contribute to the supervision.

NETWORK SUPERVISION

The network is jointly supervised by the three loss terms defined for the three separate branches. The total loss is given as follows:

$$L = L_{\mathrm{s}} + \lambda_1 L_{\mathrm{h}} + \lambda_2 L_{\mathrm{o}}, \tag{5.5}$$

where λ_1 and λ_2 denote the hyperparameters to balance the corresponding losses.

5.3.2. STEM LOCALIZATION

Having obtained the network outputs, we first select the points predicted as *stem* and perform a density-based clustering on the x-y plane over the stem points to initially localize individual tree roots. This is inspired by the observation that stem points typically form dense clusters at tree roots.

In practice, not all identified stem clusters correspond to actual tree stems due to noise and errors in the previous semantic predictions. For example, objects such as lampposts are often misclassified as tree stems. To mitigate such errors, we perform a sequence of fidelity checks over the clustered stem candidates. Algorithm 1 details our stem localization steps. Three criteria are designed based on the geometrical properties of tree roots and heat value distributions, which can be formulated as follows:

(i) **Proximity to the ground**. We measure if the lowest point in a tree stem candidate is close enough to the ground using a height threshold ϵ_z, as illustrated in Algorithm 1 Line 9. This ensures that detected stems are grounded appropriately within the scene.

(ii) **High heat value**. As illustrated in Figure 5.3, true stem points are likely to be predicted with higher heat values. Therefore, we use an empirical heat threshold ϵ_h to filter out the stem cluster candidates with insufficient heat values. This is explained in Algorithm 1, Line 10.

(iii) **Local maxima in the heat distribution**. In certain cases, such as urban scenes with small trees, true stem points may exhibit low predicted heat values as

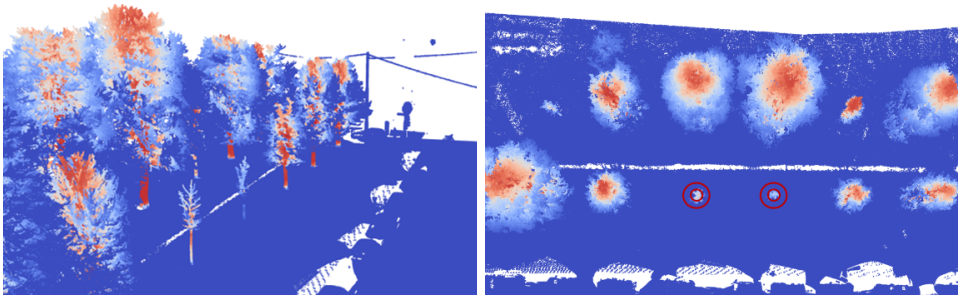
Algorithm 1: Tree stem localization from the network predictions

Input: input point coordinates P, predicted semantic map S, and heatmap H
Output: stem clusters $C = \{C_1, C_2, C_3, ...\}$ with root locations $R = \{\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, ...\}$

1 **Initialization:** $C \leftarrow \emptyset$, $R \leftarrow \emptyset$, user-defined thresholds $\epsilon_h, \epsilon_z, \epsilon_r$
2 Extract points with semantic predictions of *stem* and *crown*,
  $P_s \leftarrow \{\mathbf{p_i} \in P | s_i = stem\}$;
  $P_c \leftarrow \{\mathbf{p_i} \in P | s_i = crown\}$
3 Apply density-based clustering on P_s on the x-y plane to obtain stem cluster candidate set $C' = \{C_1, C_2, C_3, ...\}$
4 **for** $C_i \in C'$ **do**
5 Collect the stem cluster coordinates and heatmap predictions,
   $P_i \leftarrow \{\mathbf{p_j} | \mathbf{p_j} \in P_s \cap C_i\}$;
   $H_i \leftarrow \{h_j | \mathbf{p_j} \in P_s \cap C_i\}$
6 $z_{min} \leftarrow min(P_i(z)) - z_{ground}$
7 $h \leftarrow avg(H_i)$
8 $\mathbf{r_i} \leftarrow avg(P_i)$
9 **if** $z_{min} < \epsilon_z$ **then**
10 **if** $h > \epsilon_h$ **then**
11 $C \leftarrow C + \{C_i\}$; $R \leftarrow R + \{\mathbf{r_i}\}$
12 **end**
13 **else**
14 $N_1 \leftarrow \{\mathbf{p_j} | \mathbf{p_j} \in P_i \wedge dist(\mathbf{p_j}, \mathbf{r_i}) \leq \epsilon_r\}$; // $dist(\cdot,\cdot)$ computes the 2D distance of points on x-y plane
   $N_2 \leftarrow \{\mathbf{p_j} | \mathbf{p_j} \in P_i \wedge dist(\mathbf{p_j}, \mathbf{r_i}) \leq 3\epsilon_r\}$;
   $h_1 \leftarrow max\{h_j | \mathbf{p_j} \in N_1\}$; $h_2 \leftarrow max\{h_j | \mathbf{p_j} \in N_2\}$
   if $h_1 \geq h_2$ **then**
15 $C \leftarrow C + \{C_i\}$; $R \leftarrow R + \{\mathbf{r_i}\}$
16 **end**
17 **end**
18 **end**
19 **end**

the heatmap prediction is implemented as a regression task that outputs continuous predictions. Figure 5.4 shows such an example. Therefore, simple thresholding using ϵ_h will likely ignore such small trees with low heat values. To enhance the robustness of stem detection, we apply a two-layer cylindrical neighborhood, as illustrated in Figure 5.4, to assess whether a stem cluster represents a local maximum in the heat value distribution. Stem cluster candidates that form such local maxima are also classified as true tree stems, which serves as a refinement for the second criterion. Our local maxima-based stem filtering is detailed in Algorithm 1, Lines 14-16.

Although the design of the local maxima criterion is not as explicit as the first two criteria, we found that it significantly improves stem recognition in our experiments. Its effectiveness is discussed with ablation studies in Section 5.4.6.



(a) Side view (b) Top view

Figure 5.4: An example of predicted heatmaps for an urban scene. The Gaussian heat values range from 0 to 1, from blue to red. Although stem points from large trees typically have high heat values, small tree stem points exhibit low predicted heat values, making it difficult to detect them successfully. We propose to detect these stems by detecting the local maxima in the heat distribution using a two-layer cylindrical neighborhood, visualized as red circles.

5.3.3. TREE POINT GROUPING

Following the localization of tree stems, we perform individual tree isolation using a graph-based approach. We construct a Delaunay triangulation graph $G = (V, E)$ over the input tree points (i.e., points predicted to be either *stem* or *crown*). The detected tree root locations, $R = \{\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, ...\}$ obtained from Algorithm 1, are manually added to G as additional vertices. Then, we apply Dijkstra's shortest path algorithm to build the Minimum Spanning Tree (MST), determining the tree source for each vertex within the graph. We add pseudo edges between root vertices in R with zero edge weights, which guarantees that every tree vertex can be sourced back to a root vertex in the resulting MST.

Works of Livny *et al.* (2010) and Tao *et al.* (2015) also utilize the shortest path approach to group individual tree points based on the tree root locations. However, they only weigh the edges of G by the 3D Euclidean distances between vertex pairs. While straightforward, this edge-weighting strategy fails to account for size variations among different tree instances in complex forest scenes. For example, branches or twigs from a large tree may be wrongly assigned to a neighboring smaller tree if their shortest paths to the neighboring tree's root are shorter. This often leads to inaccurate boundary delineation between tree instances.

To overcome their limitations, we propose shifting the vertex coordinates using the predicted offset embeddings obtained from Section 5.3.1. Edges in G are then weighted by the 3D Euclidean distances between the shifted vertex pairs. Given the 3D coordinate of the original vertex $\mathbf{v_i}$, we shift it to a new coordinate $\mathbf{v_i'}$ by the following formula:

$$\mathbf{v}_\mathbf{i}' = \mathbf{v}_\mathbf{i} + \beta(1 - h_i)\frac{\mathbf{d}_\mathbf{i}}{\|\mathbf{d}_\mathbf{i}\|_2}, \tag{5.6}$$

where h_i and $\mathbf{d}_\mathbf{i}$ are the predicted heat value and offset vector of the i^{th} point in the graph. β is a user-defined hyperparameter. The shifting mechanism enables each tree point to move incrementally toward its corresponding instance centroid. In Equation 5.6, the step direction is determined by the predicted offset $\mathbf{d}$, and the step magnitude is modulated by the predicted heat value h. Thus, points near tree stems or main branches will hardly shift, whereas points near instance boundaries undergo more significant shifts toward the instance centroids.

Subsequently, the edge weight e_{ij} between the i^{th} vertex and the j^{th} vertex of the graph G is computed as follows:

$$e_{ij} = \|\mathbf{v}_\mathbf{i}' - \mathbf{v}_\mathbf{j}'\|_2. \tag{5.7}$$

We apply Dijkstra's shortest path algorithm to separate individual trees based on the edge weights obtained from Equation 5.7, enhancing the robustness of tree segmentation against variations in tree size and shape. In Section 5.4.6, we present ablative studies on isolating tree instances using shifted 3D coordinates.

5.4. EXPERIMENTS

5.4.1. DATASETS

We test our method on point clouds ranging from urban scenes to forest scenes. Since our method is primarily designed to segment urban trees from scene-level urban point clouds. To evaluate its performance, we use TreeML (Yazdi *et al.*, 2024), a large-scale labeled urban forest point cloud dataset that can be publicly accessed online. TreeML consists of 40 urban street scenes in Munich captured by MLS scanners and contains 3,755 trees representing a wide range of sizes and species. Each tree is measured by the Quantitative Structure Modeling approach, enabling us to extract GT tree stem points. We use 30 scenes for training, five scenes for validation, and five scenes for testing.

Meanwhile, we also assess the applicability of our approach to natural forest scenes using another dataset, ForInstance (Puliti *et al.*, 2023). ForInstance is an ALS dataset that captures dense forest areas and comprises five collections from diverse global regions (i.e., Norway, the Czech Republic, Austria, New Zealand, and Australia) representing varying forest types. It contains both instance-level annotations and part-level annotations, such as stem and branch. From the publicly available 32 forestry scenes, we use 21 scenes for training and 11 for testing, following the official benchmark's recommended split.

5.4.2. IMPLEMENTATION DETAILS AND HYPERPARAMETERS

We adopt PointMetaBase (H. Lin *et al.*, 2023), an MLP-based approach, as the point feature learning encoder, as explained in Section 5.3.1.

Since both datasets contain massive points that are challenging for the network to process directly, we crop the scanned scenes into small patches and use these as the input to our network. Additionally, PointMetaBase uses a voxel subsampling strategy to further reduce the number of points. We set the voxelization grid size as 20cm, which is appropriate for processing large-scale urban scenes (Q. Hu *et al.*, 2021). Following the original design of PointMetaBase, we employ the cosine learning rate scheduler with an initial rate of 0.01. The AdamW optimizer is used with a weight decay of 0.0001. Also, we use several data augmentation techniques, such as adding jitter noise to point coordinates, randomly rotating points, and randomly dropping point intensities, to enhance network learning. The network is trained for 50 epochs with a batch size of 16. We implement stem localization and tree point grouping in C++. A detailed summary of the hyperparameters used for network training and tree instance segmentation is given in Table 5.1.

	Eq. 5.1			Eq. 5.2	Eq. 5.5		Eq. 5.6	Alg. 1		
	w_1	w_2	w_3	α	λ_1	λ_2	β	ϵ_z	ϵ_h	ϵ_r
TreeML	2.0	15.0	1.0	10.0	10.0	0.05	3.5	2.0	0.5	0.15
ForInstance	2.0	6.0	3.0	10.0	10.0	0.2	2.5	2.0	0.9	0.1

Table 5.1: Details of network supervision and segmentation hyperparameters. Here, w_1, w_2, and w_3 represent the semantic weights of *crown*, *stem*, and *other*, respectively.

5.4.3. COMPARISON AND EVALUATION

We compare our developed method with two open-source approaches TreeSeparation (J. Wang *et al.*, 2018) and TreeLearn (Henrich *et al.*, 2024). TreeSeparation is a heuristic-based approach that takes the pure tree points as input and performs layer-wise clustering for tree instance segmentation. For this comparison, we use the semantic predictions of our approach as input to TreeSeparation. TreeLearn represents the state-of-the-art in learning-based methods. It takes the scene point clouds as input and directly generates the tree instance predictions. To ensure a fair comparison, we use the same input feature (i.e., point intensity) and voxel resolution and set the same train-test split for both our network and TreeLearn.

To quantitatively assess the tree segmentation performance, we adopt the widely used Average Precision (AP) metric, which provides a robust measure of model performance in instance segmentation tasks. AP is computed by deriving the Precision and Recall values across various confidence thresholds and calculating the area under the resulting Precision-Recall (PR) curve. Let TP, TN, FP, and FN

represent the number of true positives, true negatives, false positives, and false negatives, respectively. The Precision and Recall metrics are formally defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{5.8}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{5.9}$$

Precision quantifies the proportion of predicted positive instances that are correctly classified, whereas Recall measures the proportion of actual positive instances that are accurately identified by the model. Both metrics are inherently dependent on a predefined confidence threshold. To mitigate this dependency and provide a threshold-independent evaluation, the AP metric summarizes model performance by approximating the area under the PR curve. The general formulation of AP is defined as follows:

$$AP = \int_0^1 p(r)\, dr, \tag{5.10}$$

where r denotes the Recall value and $p(r)$ is the corresponding Precision at that Recall level. In practice, the integral is replaced with a finite summation over the set of discrete, unique Recall values obtained across a range of confidence thresholds, i.e.,

$$AP = \sum_i (r_i - r_{i-1}) \cdot p_i, \tag{5.11}$$

where p_i and r_i are the Precision and Recall at the i^{th} threshold.

Following the common practice (L. Jiang *et al.*, 2020) in 3D instance segmentation tasks, we report AP scores using IoU thresholds of 25% and 50%, denoted as AP_{25} and AP_{50}. We also report the overall AP score averaged across IoU thresholds ranging from 50% to 95% with a step length of 5%.

5.4.4. RESULTS OF URBAN FORESTRY SCENES

In Table 5.2, we present our instance segmentation results achieved on TreeML (Yazdi *et al.*, 2024) dataset in comparison with TreeSeparation (J. Wang *et al.*, 2018) and TreeLearn (Henrich *et al.*, 2024). Our proposed SATree achieves the highest scores across all five test scenes as evaluated by the AP, AP_{50}, and AP_{25}, surpassing the comparison methods by a large margin. In particular, SATree attains an AP score exceeding 0.9 in four out of the five scenes. This shows that our method delivers promising instance segmentation of trees in most urban scenes, highlighting that it is consistently practical for the tree instance segmentation task in urban environments.

Scene street name	metric	TreeSeparation	TreeLearn	SATree (ours)
2023-01-09_tum_campus	AP	0.697	0.685	**0.935**
	AP_{50}	0.815	0.800	**0.969**
	AP_{25}	0.846	0.845	**0.985**
2023-01-10_47	AP	0.770	0.726	**0.869**
	AP_{50}	0.821	0.786	**0.893**
	AP_{25}	0.927	0.784	**0.964**
2023-01-12_57	AP	0.905	0.845	**0.972**
	AP_{50}	0.964	0.929	**1.000**
	AP_{25}	0.964	0.929	**1.000**
2023-01-13_70	AP	0.810	0.684	**0.918**
	AP_{50}	0.924	0.771	**0.983**
	AP_{25}	0.932	0.780	**0.983**
2023-01-16_44	AP	0.665	0.851	**0.981**
	AP_{50}	0.875	0.918	**0.997**
	AP_{25}	0.944	0.936	**1.000**

Table 5.2: Tree instance segmentation results on TreeML (Yazdi *et al.*, 2024) with AP, AP_{50}, and AP_{25}. Compared to the other two methods, i.e., TreeSeparation (J. Wang *et al.*, 2018) and TreeLearn (Henrich *et al.*, 2024), SATree achieves the highest scores (in bold), outperforming on all scenes by a large margin.

Figure 5.5 visually compares the segmentation results achieved by TreeSeparation, TreeLearn, and our method. TreeML captures complex urban street-level scenes exhibiting massive trees with varying sizes and shapes, which poses significant challenges for existing methods, resulting in under-segmentation or over-segmentation. For example, in Figure 5.5 (row 1), three trees with closely intertwined branches and twigs are erroneously segmented as a single tree by TreeSeparation and TreeLearn. In contrast, our proposed SATree successfully identifies and segments the three individual trees. Meanwhile, over-segmentation, where a single tree is divided into multiple smaller segments, is also common due to the large volume of tree canopies, which can be observed in rows 2, 4, and 5 of Figure 5.5. Unlike TreeSeparation and TreeLearn, SATree mitigates most over-segmentation errors by leveraging its robust stem localization strategy.

We also observe that our method can detect trees of petite sizes, which are often overlooked by the comparative method TreeLearn (see Figure 5.5 row 3). In TreeLearn, trees smaller than 10 meters are likely to be misclassified as background points due to the sparsity of small tree representations in the training dataset. However, despite using the same training data, SATree can still effectively identify small trees. This is attributed to SATree's ability to accurately detect the roots of small trees, which facilitates follow-up tree instance segmentation. Moreover, our approach

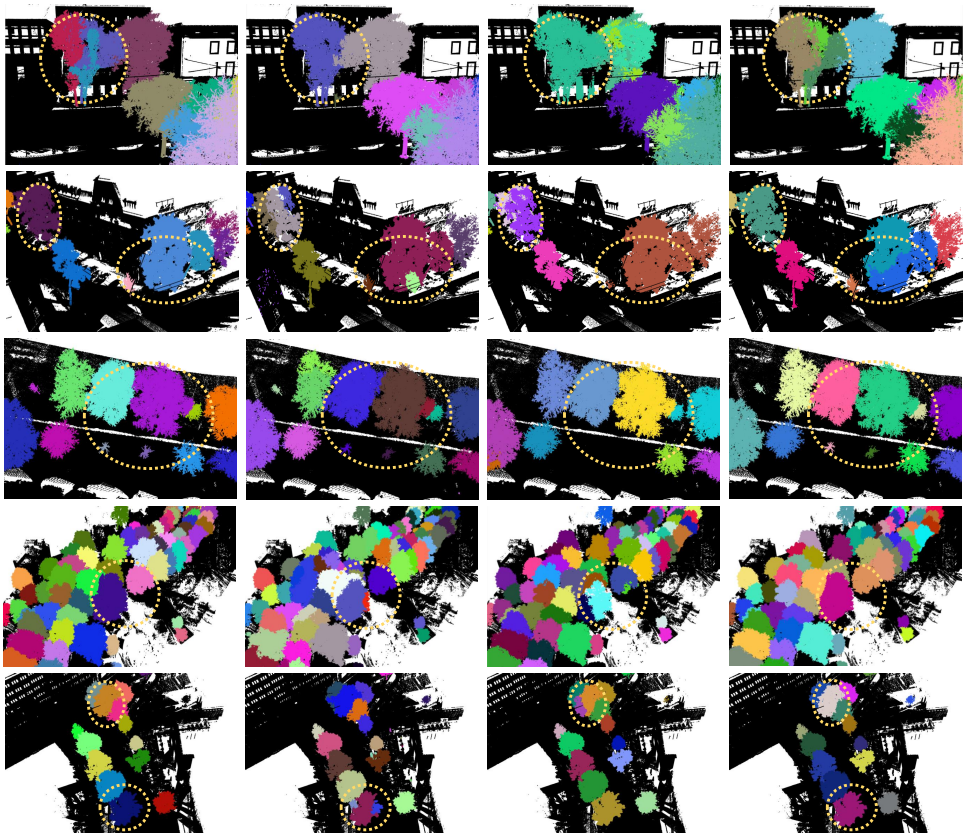


Figure 5.5: Tree instance segmentation results achieved on the TreeML dataset by applying TreeSeparation (J. Wang *et al.*, 2018), TreeLearn (Henrich *et al.*, 2024), and our method, respectively. The segmented tree instances are randomly colored, and the background points are shown in black.

can also be directly integrated with TreeSeparation. Using the semantic predictions of our approach as input, TreeSeparation also successfully identifies small trees.

5.4.5. RESULTS OF NATURE FORESTRY SCENES

Our approach is primarily designed to address the task of urban tree instance segmentation. To further assess its applicability to natural forestry environments, we also evaluate it on the ForInstance (Puliti *et al.*, 2023) dataset.

Table 5.3 reports our performance scores compared against TreeSeparation (J. Wang *et al.*, 2018) and TreeLearn (Henrich *et al.*, 2024). Overall, our SATree outperforms the other two methods in most scenes, achieving the highest AP, AP_{50}, and AP_{25} scores in

Forest scene name	metric	TreeSeparation	TreeLearn	SATree (ours)
CULS	AP	0.922	0.839	**1.000**
	AP_{50}	1.000	0.900	**1.000**
	AP_{25}	1.000	0.900	**1.000**
NIBIO	AP	0.456	0.647	**0.665**
	AP_{50}	0.604	0.764	**0.814**
	AP_{25}	0.629	0.783	**0.857**
RMIT	AP	0.348	0.153	**0.366**
	AP_{50}	0.500	0.230	**0.541**
	AP_{25}	0.616	0.343	**0.670**
SCION	AP	0.354	**0.822**	0.788
	AP_{50}	0.690	0.884	**0.907**
	AP_{25}	0.857	0.884	**0.930**
TUWIEN	AP	0.162	**0.349**	0.295
	AP_{50}	0.343	**0.514**	0.486
	AP_{25}	0.457	0.627	**0.743**

Table 5.3: 3D instance segmentation results on ForInstance (Puliti *et al.*, 2023) with AP, AP_{50}, and AP_{25} scores. Compared to the other two methods, i.e., TreeSeparation (J. Wang *et al.*, 2018) and TreeLearn (Henrich *et al.*, 2024), SATree achieves the highest scores in three of the five forest scenes.

the forest scenes CULS, NIBIO, and RMIT. Notably, SATree achieves 100% precision in instance segmentation for the CULS scene. For the SCION and TUWIEN forest scenes, TreeLearn achieves the highest overall AP score. Nevertheless, the proposed SATree outperforms TreeLearn in AP_{50} and AP_{25}, demonstrating its robustness across different evaluation metrics.

Figure 5.6 visually compares the tree instance segmentation results achieved by TreeSeparation, TreeLearn, and our proposed SATree. As in TreeML, SATree reduces under-segmentation and over-segmentation errors, outperforming the other two methods. In the CULS scene (Figure 5.6, the first row), SATree avoids segmenting a single tree crown into multiple sub-trees. In the scenes of NIBIO (Figure 5.6 row 2) and RMIT (Figure 5.6 row 3), SATree successfully detects the trees ignored by other methods, demonstrating the robustness of its stem localization strategy. For the TUWIEN (Figure 5.6 row 5) scene, SATree generates segmentation outputs closely aligned with the GT. However, despite this visual alignment, the reported AP scores of SATree are lower than those of TreeLearn. This discrepancy arises because our method segments bushes as individual trees, while GT labels annotate them as background points. This results in more false positives and thus leads to a decrease in the final AP score.

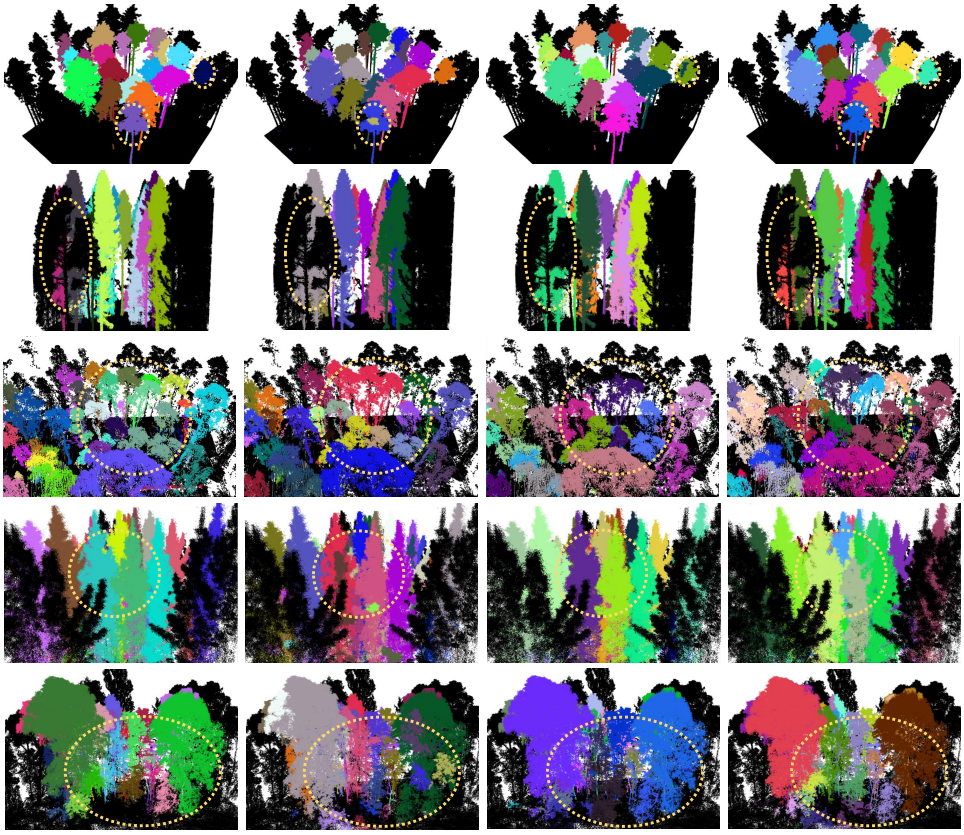


Figure 5.6: Tree instance segmentation results achieved on the ForInstance dataset by applying TreeSeparation (J. Wang *et al.*, 2018), TreeLearn (Henrich *et al.*, 2024), and SATree, respectively. The five forest scenes (CULS, NIBIO, RMIT, SCION, and TUWIEN) are shown from top to bottom. We use randomized colors to visualize segmented tree instances, while background points are shown in black.

In SCION, SATree does not perform as well as TreeLearn in terms of both quantitative measures (Table 5.3) and qualitative results (Figure 5.6 row 4). It is observed that for coniferous trees, although SATree can correctly localize individual trees by detecting their stems, it struggles to delineate tree instance boundaries when they are too closely spaced. This is because the adopted shortest-path algorithm struggles to separate closely adjacent trees, even with guidance from the learned offset embeddings. Therefore, the resulting tree segmentation boundaries are noisier than TreeLearn's. Overall, our method achieves comparable or superior results to other existing methods specifically designed for natural forest scenes, indicating its potential for such application scenarios.

5.4.6. ABLATION STUDIES

In our pipeline, we introduce several novel strategies to enhance accurate tree instance segmentation: (1) Predicting a Gaussian heatmap in the network that benefits stem extraction, as illustrated in Section 5.3.1; (2) Detecting local maxima from the heat distribution to robustly recognize tree stems from cluster candidates, as explained in Section 5.3.2; (3) Grouping tree points into distinct instances using shifted coordinates instead of raw coordinates to better delineate tree boundaries, as detailed in Section 5.3.3 (Equation 5.6).

We perform ablative studies to verify the effectiveness of the proposed strategies. We use the *2023-01-09_tum_campus* test scene from TreeML for our ablation studies, given its challenges including dense tree overlaps and significant variations in tree sizes. Thus, performance in this scene provides a substantial measure of our proposed strategies. Table 5.4 summarizes the results.

Test scene	metric	w/o heatmap	w/o local maxima	w/o shifting	SATree
2023-01-09_tum_campus	AP	0.831	0.892	0.713	**0.935**
	AP_{50}	0.892	0.923	0.862	**0.969**
	AP_{25}	0.938	0.938	0.969	**0.985**

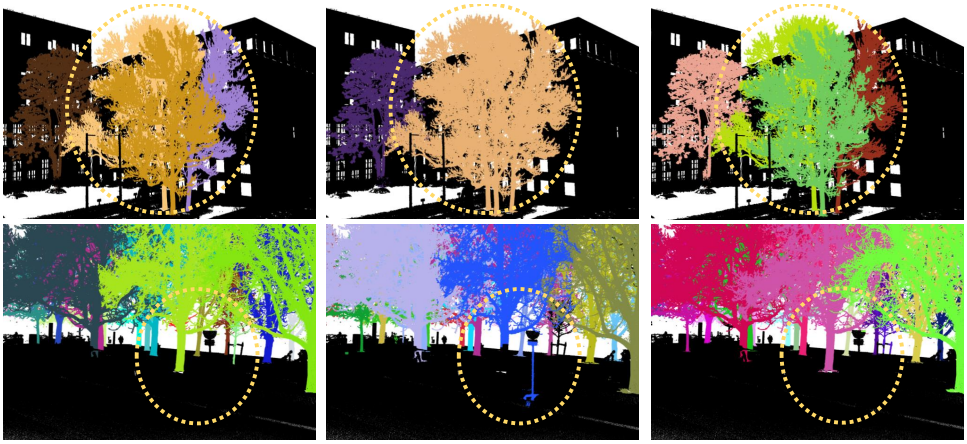
Table 5.4: Ablative results achieved by omitting the following key components: heatmap prediction (Section 5.3.1), local maxima identification for stem extraction (Section 5.3.2), and using shifted coordinates for tree point grouping (Section 5.3.3, Equation 5.6).

HEATMAP PREDICTION

To assess the effectiveness of heatmap prediction for tree instance segmentation, we remove this branch from the network along with the supervision term L_h in Equation 5.5. Additionally, we remove the heat value-related criteria from Algorithm 1 and solely rely on the geometrical criterion for stem extraction. In the absence of heat value predictions, Equation 5.6 is modified accordingly to

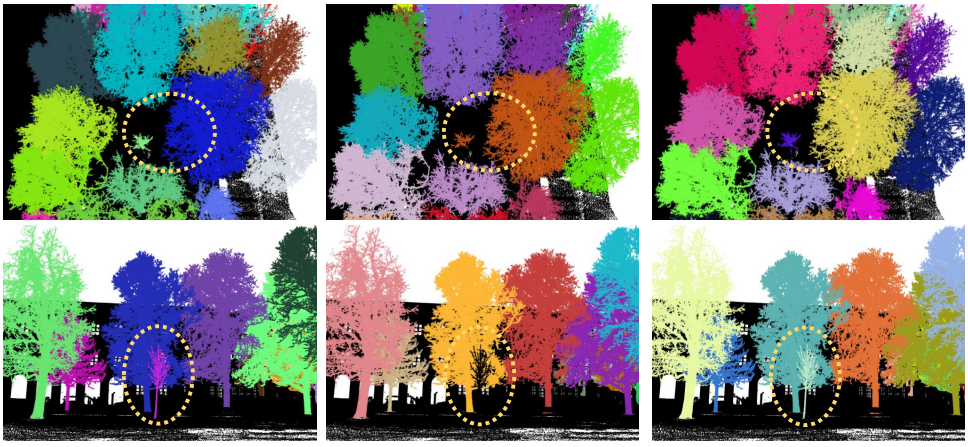
$$\mathbf{v_i'} = \mathbf{v_i} + \mathbf{d_i}, \tag{5.12}$$

where we use the magnitude of the predicted offset vector $\mathbf{d}$ to determine the extent of shifting. As shown in Table 5.4, adding the heatmap prediction task results in improved performance, achieving an increase of 0.104 in AP, 0.077 in AP_{50}, and 0.047 in AP_{25}. Figure 5.7 presents the visual comparison between the results obtained without and with the heatmap prediction branch. These results suggest that heatmap prediction enhances stem localization accuracy and yields cleaner instance segmentation of trees.



GT instances without heatmap prediction with heatmap prediction

Figure 5.7: Visual comparison of the results achieved with and without adopting the heatmap prediction task in the network.



GT instances without local maxima detection with local maxima detection

Figure 5.8: Visual comparison of the results achieved with and without using the local maxima of the heatmap distribution for stem identification.

STEM EXTRACTION BY DETECTING LOCAL MAXIMA

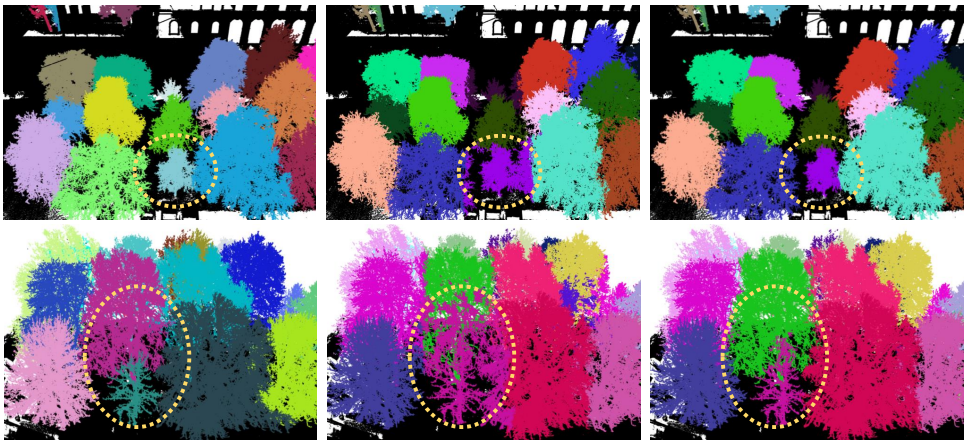
In Section 5.3.2, we refine the stem localization by detecting whether a stem candidate location is a local maximum within the heat distribution. To verify the effectiveness of this strategy, we remove the detection of local maxima and instead only use the heat value threshold to select stems from cluster candidates. Table 5.4 shows that without the local maxima detection, performance scores drop 0.043 in AP, 0.046 in AP_{50}, and 0.047 in AP_{25}. The main reason is that detecting local maxima

helps to recognize small trees with thin stem structures, which are often overlooked when using heat value thresholds alone.

Figure 5.8 illustrates that the absence of local maxima detection leads to poor segmentation of small trees, either merging them with nearby trees or labeling them as background noise. In contrast, including the local maxima detection significantly enhances performance by reliably identifying most of the smaller trees.

TREE POINT GROUPING WITH SHIFTED COORDINATES

Constructing a forest graph and applying the shortest-path algorithm to isolate single trees have been explored in a few studies (Livny *et al.*, 2010; Tao *et al.*, 2015). Nevertheless, these methods primarily assign tree points to instance roots based on their spatial proxies in the 3D Euclidean space. Different from that, we shift the raw coordinates of 3D tree points using the learned offset embeddings and segment tree instances in the shifted 3D space.



GT instances | raw coordinate grouping | shifted coordinate grouping

Figure 5.9: Visual comparison of the results achieved using raw coordinates or using shifted coordinates in the tree point grouping.

In Table 5.4, we compare the results obtained by grouping tree points based on their raw coordinates versus the results achieved using shifted coordinates. Without using shifted coordinates for tree point grouping, performance scores decrease 0.222 in AP, 0.107 in AP_{50}, and 0.016 in AP_{25}. Specifically, compared to the other ablation experiments, this configuration achieves a higher AP_{25} score but lower scores in AP_{50} and AP. The reason is that grouping tree points using raw coordinates does not impact the stem extraction process. Tree stems could still be accurately localized, leading to a high AP_{25} score. However, it fails to segment regions near tree boundaries, particularly in scenarios with significant overlap between tree branches, thus resulting in poorer performances in AP and AP_{50}. The visual comparison

presented in Figure 5.9 shows that grouping tree points with raw coordinates fails to preserve tree boundaries, especially when nearby trees exhibit varying sizes. As a result, points from an enormous tree are likely to be assigned to a smaller neighboring tree. In contrast, grouping tree points with shifted coordinates significantly prevents such errors and obtains more natural tree instance boundaries.

5.4.7. LIMITATIONS

Our SATree demonstrates high-fidelity tree instance segmentation in complex urban environments by leveraging detected tree structures, such as stems. However, it still suffers from three primary limitations.

First, the proposed SATree assumes that tree stems are captured with high data quality. When significant portions of stems are missing or trunk points are excessively sparse, our method may fail to segment tree instances. Second, SATree may encounter difficulties in accurately delineating tree boundaries for certain forest types such as coniferous forests, as illustrated in Figure 5.6 (row 4). This is because the adopted shortest-path algorithm may struggle to separate densely connected coniferous trees. Last, our method is not designed for end-to-end training. The second stage of the framework, tree point grouping, lacks direct supervision with GT tree instance labels. Integrating a trainable module for tree instance isolation could enhance segmentation performance by leveraging additional network supervision.

5.4.8. POTENTIAL APPLICATIONS

Our method produces accurately segmented 3D tree instances in urban environments, which are beneficial for various forestry-related applications such as 3D vegetation reconstruction, measuring tree heights and crown volumes, biomass estimation, and more. In addition, our method also learns to generate Gaussian heatmaps for urban trees, where high heat values indicate tree stems or main branch structures and low heat values are associated with small twigs near tree instance boundaries. This characteristic of the heatmaps suggests their potential utility in part-level segmentation of trees. Another by-product of our method is the generated MST graphs in the tree point grouping process, as explained in Section 5.3.3. These MSTs approximate the skeletal structures of the tree instances and can offer valuable information for reconstructing 3D forest structures from raw input data.

5.5. CONCLUSIONS

In this chapter, we have proposed a novel deep learning-based framework for instance segmentation of trees from LiDAR point clouds. The core innovation of our method lies in leveraging key structural features, such as stems and crowns, to segment individual trees. To achieve this, we developed a multi-task learning network that simultaneously classifies tree crown points and tree stem points.

Additionally, we introduced heatmap prediction and offset prediction tasks to guide the tree instance segmentation. Key strategies, such as localizing stems by detecting local maxima in the heatmap and grouping tree points via shifted coordinates, proved effective for accurate tree segmentation. Extensive experiments on two public forestry datasets demonstrated the superiority of our method over state-of-the-art methods. Notably, our method showed strong performance in segmenting trees in large urban street-level scenes and generalized well to natural forest environments, achieving comparable or superior results compared to existing approaches.

For recommendations of future work, researchers may consider integrating the instance segmentation module as a trainable component, enabling end-to-end supervision. Additionally, developing an adaptive strategy to automatically determine optimal hyperparameters for specific datasets can also enhance the usability and versatility of the proposed SATree approach.

6

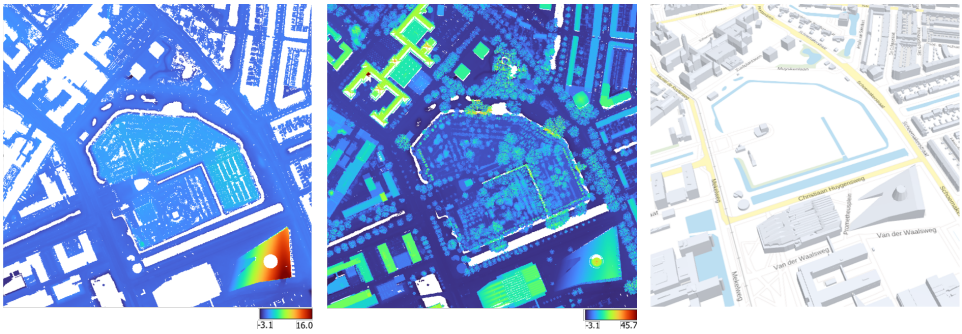
CONFIDENCE-BASED ONLINE LEARNING FOR REAL-WORLD POINT CLOUDS

From Chapter 3 to Chapter 5, we have explored fully supervised methods for 3D semantic segmentation and instance segmentation from point clouds. Despite advancements in these methods, accurately processing and interpreting real-world datasets remains a critical challenge due to inevitable data outliers, uncertainties, and annotation errors. This chapter investigates a confidence-based deep learning framework to improve the classification accuracy of real-world point cloud data. By incorporating multi-source information, such as aerial imagery, and embedding geospatial prior knowledge, this framework models data uncertainty through point-wise confidence scores, which allows the network to refine both its predictions and the quality of training labels via iterative online learning. Extensive experiments on large-scale airborne LiDAR data, i.e., Dutch AHN dataset (AHN, 2025), demonstrate that the proposed method effectively enhances training data by reducing label noise and improving annotation quality, thereby leading to more robust and generalizable model performance.

This chapter is based on the MSc Thesis in Geomatics at the TU Delft titled "A Confidence-aware Deep Learning Framework for Refining Laser-scanned Point Cloud Classification" by Madanu (2024) and the paper: Sharath Chandra Madanu*, Shenglan Du*, Jantien Stoter, Daan van der Heide. RefineNet: a Confidence-aware Deep Online Learning Framework to Refine Real-world Point Cloud Semantic Segmentation. 2026 ISPRS Congress. * denotes equal contribution. As the first supervisor of the MSc thesis, I contributed to identifying the research problem, designing the methodology, providing code support, and writing.

6.1. INTRODUCTION

Point cloud data has become a critical source of geospatial information, supporting a wide range of downstream applications, as discussed in previous chapters. Analyzing and interpreting such 3D data is increasingly driving innovation not only in academic research, but also in industry practice (Biljecki *et al.*, 2015). A prominent example is the Actueel Hoogtebestand Nederland (AHN) dataset (AHN, 2025), the nationwide airborne LiDAR point cloud resource that captures high-resolution elevation and topographic information across the Netherlands. It has been regularly updated and extensively utilized in applications such as the creation of DTM and DSM models, urban structure reconstruction, and hydrological and water resource management. Figure 6.1 presents representative examples of DSMs, DTMs, and reconstructed 3D building models derived from AHN point cloud data.



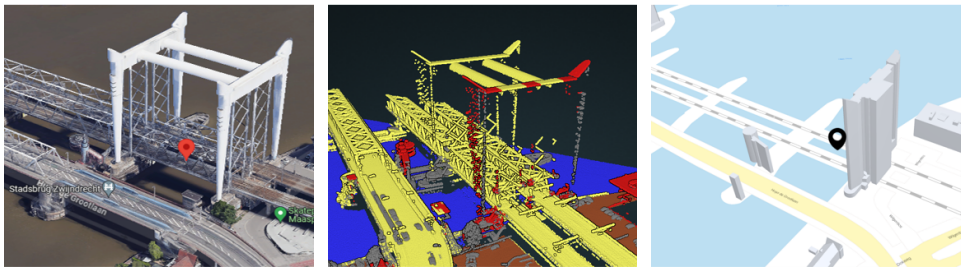
(a) DTM (b) DSM (c) 3D BAG

Figure 6.1: Representative downstream applications of ALS point clouds.[1]

However, unlike the research-oriented datasets used in previous Chapters 3 and 4 such as S3DIS (Armeni *et al.*, 2016) and SensatUrban (Q. Hu *et al.*, 2021), real-world point cloud datasets usually exhibit inherent data quality issues, including noise, outliers, and annotation inconsistencies. Such errors can compromise the reliability of data interpretation and substantially degrade the quality of derived geospatial products. Moreover, such errors can propagate through subsequent analyses, potentially resulting in misleading outcomes in applications such as urban fluid dynamics and hydrological modeling. For instance, misclassification of *ground* points as part of *building* structures can distort the generation of DTM maps, leading to erroneous ground surface representations within building footprints, as demonstrated in Figure 6.2.

Semantic segmentation of point clouds initially relied on manual annotation and human supervision. Various AI-based techniques have been developed to automate this process. In particular, deep learning methods have demonstrated significant

[1]Images are sourced from online.
DTM and DSM: https://viewer.ahn.nl/AHN4/DTM/0/6.5465/52.26738/3
3D BAG: https://3dbag.nl/en/viewer



(a) Colored point clouds (b) ALS misclassification (c) Reconstruction results

Figure 6.2: Misclassification of bridge points as building points leads to urban reconstruction errors. These colors are used to render the ALS point cloud classification code: ■ for *building*, ■ for *civil structure*, ■ for *water*, and ■ for *others*.

advantages over classical machine learning methods by eliminating the need for manual feature engineering. Chapter 2 has provided a comprehensive overview of deep learning-based methods for point cloud classification and segmentation, including network architectures based on MLPs, convolutions, and transformers.

Despite these advancements, deep learning models remain vulnerable to data quality issues, with model performance heavily dependent on the quality of the input data. High-quality inputs generally lead to accurate predictions, whereas noisy data can substantially affect or downgrade the performance. Although several data-efficient strategies have been introduced to address data quality and scarcity issues, many rely on complex architectures, such as Generative Adversarial Networks (GANs) (H. Li *et al.*, 2021), and require extensive training procedures. This underscores the need for a more straightforward yet practical approach to enhance model performance, particularly in correcting misclassifications within point cloud data.

This chapter proposes an online learning framework that integrates a confidence-based training label updating strategy to refine point cloud classification on real-world datasets with low-quality annotations. By incorporating geospatial knowledge as priors, our method assigns a point-wise confidence score to assess the reliability of each training point label. These scores are then used to dynamically guide the training process: high-confidence samples are prioritized for learning, while low-confidence samples are iteratively refined. This strategy enhances both the quality of the training data and the segmentation performance of the model. Overall, the proposed approach aims to address the limitations of existing techniques by providing a straightforward yet effective solution that improves segmentation accuracy without the increased computational complexity associated with more sophisticated architectures.

We summarize our key contributions as follows:

- We introduce a confidence estimation strategy that assesses the reliability of existing semantic labels by leveraging local semantic consistency and incorporating geometric priors from supplementary data sources such as aerial imagery.
- We propose an online learning framework integrated with a label refinement mechanism, which dynamically selects high-confidence samples for training and iteratively updates labels of low-quality samples. This process enhances both the performance and robustness of the deep learning model.

6.2. RELATED WORK

While many deep learning approaches for 3D point cloud classification and segmentation are fully supervised and depend on high-quality annotated data, this section focuses on data-efficient techniques designed to cope with data with limited or low-quality annotations.

The first research direction focuses on transfer learning, where knowledge acquired from a source dataset is leveraged to enhance classification or recognition performance in a target domain. Tobin *et al.* (2017) proposed a domain randomization technique to transfer the knowledge learned from 2D simulated images to real-world object detection tasks. This *simulation-to-reality* strategy was later extended to 3D point cloud data (C. Wu *et al.*, 2023), where synthetic point clouds are generated to improve segmentation performance on real-world 3D scenes. Similarly, Xiao *et al.* (2022) introduced SynLiDAR, a large-scale synthetic LiDAR dataset collected from diverse virtual environments with varying scene configurations. Then, a point-cloud translator was developed to mitigate the domain discrepancy between synthetic and real-world data. More recently, Biehler *et al.* (2023) proposed PLURAL, a co-training framework for point cloud transfer learning that leverages contrastive instance alignment and extensive data augmentations to bridge domain gaps and improve generalization.

The second line of research is semi- and weakly supervised learning, which seeks to perform 3D semantic segmentation using fewer point labels to train deep neural networks. Wei *et al.* (2020) introduced a multi-path region mining strategy combined with the class activation mapping technique (B. Zhou *et al.*, 2016) to generate pseudo point-level labels, which are subsequently used to train the point cloud segmentation network in a fully supervised manner. Building on the idea of enhancing sparse annotations, Q. Hu *et al.* (2022) proposed the Semantic Query Network, which implicitly augments sparse supervision signals by querying and summarizing features from neighboring points, based on the assumption that nearby points share similar semantic information. In contrast, Pan *et al.* (2024) developed a label recommendation network that explicitly learns to provide recommendations for points to be labeled and perform point cloud semantic segmentation. Meanwhile, several studies have explored using self-supervised pre-training techniques to fine-tune networks on the target 3D dataset with limited annotations (Hou *et al.*,

2021; C. Sharma & Kaul, 2020; Z. Zhang *et al.*, 2021).

Furthermore, under the domain of semi-supervised learning, several research works have investigated strategies such as active learning (Settles, 2012) and self-training (Amini *et al.*, 2025) to effectively utilize both labeled and unlabeled data for enhanced model performance. Shi *et al.* (2021) introduced an active learning method based on superpoint set selection to optimize the model performance given limited annotation budgets. In a complementary direction, P. Wang and Yao (2022) designed an online pseudo-labeling framework in combination with a semantic consistency constraint, which provides additional supervisory signals to improve the robustness of point cloud segmentation under incomplete labels. H. Li *et al.* (2021) also employed unlabeled point samples and a pseudo-labeling mechanism for training. Nevertheless, this approach incorporates and trains a separate GAN architecture to pick more reliable label predictions from unlabeled point clouds.

Our approach shares conceptual similarities with the work of H. Li *et al.* (2021), which also incorporates pseudo-labeling and online learning techniques. However, our method differs in two key aspects. First, we explicitly leverage geospatial knowledge priors derived from auxiliary data sources, i.e., aerial imagery, to efficiently assess the reliability of training point labels. Second, our framework is unified, lightweight, and end-to-end trainable, without requiring the integration of multiple networks and extensive training cycles.

6.3. METHOD

Our objective is to develop an online learning framework tailored to real-world point cloud datasets, which often contain noise, outliers, and annotation artifacts. By integrating heuristic priors such as local semantic consistency and geospatial context, we estimate point-wise confidence scores that assess the reliability of individual training labels. These confidence scores are then utilized to guide the learning process, enabling the network to prioritize more trustworthy labels while simultaneously refining low-confidence annotations. Furthermore, due to the significant imbalance in point sample distributions across categories in real-world scenes, we propose to use a class-balanced loss function to supervise the network.

6.3.1. CONFIDENCE MEASUREMENT

Confidence quantifies the reliability of a point's label. It ranges from zero to one, with lower values suggesting reduced trust in the correctness of the current classification and higher values representing greater certainty. However, it is essential to note that these point-wise confidence scores provide only an approximate indication of label reliability and should not be regarded as definitive measures of labeling accuracy.

We measure confidence levels of training point labels using a two-step process:

- Initial confidence scores are assigned to each point based on local semantic

consistency within its spherical neighborhood;

- These scores are subsequently refined using auxiliary geospatial priors, including building footprint data and vegetation indices such as Normalized Difference Vegetation Index (NDVI) (DeFries & Townshend, 1994), derived from aerial imagery and DSM maps.

LOCAL SEMANTIC CONSISTENCY

Local semantic consistency measures the percentage of neighboring points that share the same semantic label as a given point, assuming that spatially proximate points tend to exhibit similar semantic characteristics. For each point i, we locate its spherical neighborhood within a radius of r. The local semantic consistency is then computed as an initial confidence score c_i, according to the following formula:

$$c_i = \begin{cases} \frac{N_j}{N_i} & \text{if } N_i \geq N_{min} \\ 0 & \text{otherwise} \end{cases}, \tag{6.1}$$

where N_i is the total number of neighboring points within the spherical neighborhood of the i^{th} point, and N_j represents the number of neighboring points that share the same classification label as the point i.

A user-defined threshold N_{min} is introduced to ensure a minimum neighborhood density, thereby reducing the influence of outliers by assigning a confidence score of zero to points with insufficient local support. We empirically set $N_{min} = 5$ in all experiments. As discussed in Section 6.4.6, we evaluate the impact of varying N_{min} and find that a threshold of 5 achieves an effective trade-off between neighborhood density and robustness to noise, minimizing the risk of assigning high confidence to sparsely supported or potentially mislabeled points.

GEOSPATIAL PRIORS

With the availability of auxiliary data sources, such as aerial images, and the use of geospatial knowledge priors, we can further refine point-wise confidence scores for specific urban object categories.

This section focuses on urban buildings, which constitute a dominant and structurally significant class in urban scene environments. Moreover, in ALS datasets, building facades and walls often exhibit sparse point densities due to sensor limitations, including flight altitude and scanner orientation. As a result, points on building facades and walls tend to have low confidence scores. Given that facades and walls are key components of a building's structural representation, it is essential to improve the confidence estimation for these regions.

Our first step is to extract building footprints from open-source DSMs and aerial ortho-imagery. Then, by projecting 3D points onto the obtained building footprints,

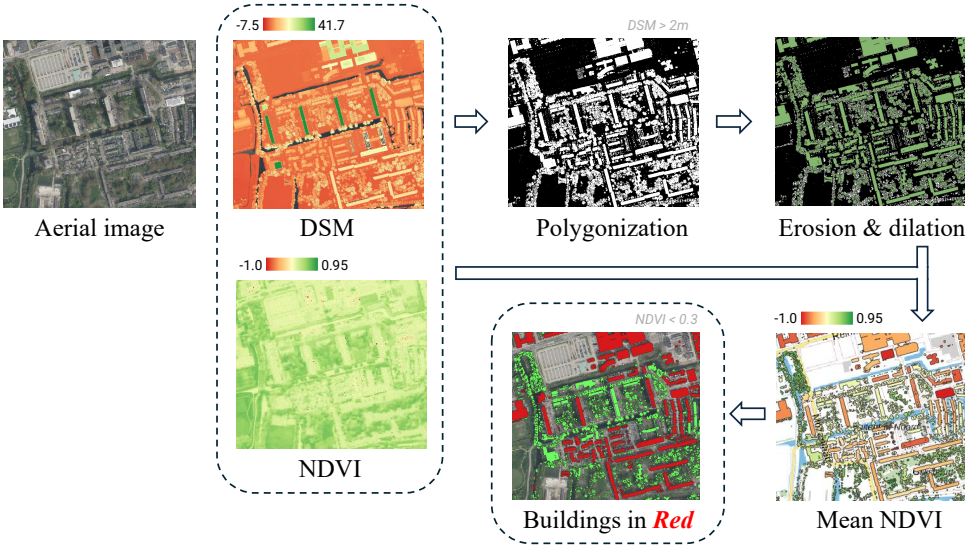


Figure 6.3: Buildings footprint extraction from open-source aerial imagery and DSMs. The mean NDVI is computed as the average NDVI value of all pixels within each extracted polygonal footprint.

confidence scores of the relevant building points, particularly those on facades, can be enhanced. Figure 6.3 illustrates the building footprint extraction process.

Our building footprint extraction procedure is outlined as follows:

1. **Height filtering and polygonization**. We binarize the input DSM map based on a height threshold ϵ_h, where pixels with elevation above ϵ_h are assigned a value of 1 and others are set to 0. The resulting binary mask is then converted into polygonal shapes, mostly containing elevated urban structures such as buildings and trees.
2. **Erosion and dilation**. We apply morphological operations, including erosion and dilation, to smooth polygon boundaries and remove noise.
3. **NDVI filtering**. NDVI is a well-established indicator of vegetative cover, which we can use to distinguish buildings from trees. We derive the NDVI map from the RGB aerial imagery following DeFries and Townshend (1994). For each candidate polygon, the mean NDVI is computed by averaging the NDVI values of all its enclosed pixels. Polygons with mean NDVI values below a predefined threshold ϵ_v are considered building footprints.

Having obtained the building footprint data, we project 3D points onto 2D building footprint polygons, enhancing the confidence scores of points corresponding to buildings to 1.0, emphasizing facade regions.

An alternative strategy is to use third-party building footprints resources, such as Basisregistratie Adressen en Gebouwen (BAG) data (BAG, 2025), which are also derived from AHN4 point clouds. However, experimental results suggest that our extraction method achieves comparable or even superior performance. As illustrated in Section 6.4.7, in some areas our extracted footprints can capture buildings that are missed in 2D BAG due to annotation errors of the point cloud data.

6.3.2. ONLINE LEARNING

The input to our network is a LiDAR point cloud, which may contain label annotation errors, along with pre-computed point-wise confidence scores. For point feature learning, we adopt KP-Conv (Thomas *et al.*, 2019) as the backbone network. While the proposed approach is compatible with various backbone architectures such as MLP-based networks (H. Lin *et al.*, 2023; Qian *et al.*, 2022) and transformer-based networks (Lai *et al.*, 2022; X. Wu *et al.*, 2024; H. Zhao *et al.*, 2021), we opt for KP-Conv given its balance between computational efficiency and representational effectiveness. Our primary focus is to investigate the impact of online learning strategies on refining real-world point cloud classification, rather than to identify the optimal deep learning architecture.

We start the network training using only point samples with high confidence scores. Following this initial training phase, the model is then used to make predictions for all points within the point cloud, resulting in per-point class probability estimates. Predictions with high probabilities overwrite the original annotation labels, forming new labels referred to as *pseudo-labels*. These pseudo-labels are integrated into the training set for model training in the next iteration. Given input data with outliers and low-quality annotations, the proposed online learning strategy enables the network to iteratively correct mislabeled samples and simultaneously improve the point-wise confidence scores. This approach yields two primary outcomes: (i) a refined and cleaned point cloud dataset with a significantly reduced number of low-confidence samples; (ii) a robustly trained network capable of accurate semantic classification, even in the presence of noisy or unreliable labels. The detailed steps of the online learning process are presented in Algorithm 2.

Our online learning strategy shares some similarities with the self-training approach proposed by H. Li *et al.* (2021). In their study, a limited set of labeled training data is augmented through GANs, which involves extensively training two networks, i.e., a segmentation network for predicting point semantics and a discriminator network assessing the quality of those predictions. While effective, such a framework can be computationally intensive. In contrast, our method employs a single network and leverages prior geospatial knowledge during preprocessing to generate point-wise confidence scores. This design choice eliminates the need for extra network components and additional training, thereby reducing computational costs and resulting in a more efficient and lightweight training pipeline.

Algorithm 2: Online learning on point clouds

Input: input point cloud X and confidence map C
Output: a trained network $f^{(e)}$ and an updated dataset $X_{\overline{U}}$

Initialization: $e \leftarrow 0$, $X_{\overline{U}} \leftarrow \emptyset$, user-defined thresholds c_1, c_2, e_1, e_2, $c_1 < c_2$, $e_1 < e_2$
Segragate X into over- and under- confident point sets X_O and X_U,
$X_O \leftarrow \{\mathbf{x_i} \in X \mid c_i \geq c_1\}$
$X_U \leftarrow \{\mathbf{x_i} \in X \mid c_i < c_1\}$
repeat
  Train $f^{(e)}$ on $X_O \cup X_{\overline{U}}$
  if $e \geq e_1$ **then**
    $\Pi_e \leftarrow \{\mathbf{x_i} \in X_U, \Phi_y(\mathbf{x_i}, f^{(e)})\}$ // $\Phi_y(\cdot,\cdot)$ denotes pseudo-labeling
    For Π_e, obtain the network softmax probability map P
    $X_e \leftarrow \{\mathbf{x_i} \mid (\mathbf{x_i}, \widetilde{y}_i) \in \Pi_e \wedge p_i \geq c_2\}$
    $X_{\overline{U}} \leftarrow X_{\overline{U}} \cup X_e$
    $X_U \leftarrow X_U \setminus X_e$
    $e \leftarrow e + 1$
  end
until $e \geq e_2$ *or* $X_U = \emptyset$;

6.3.3. CLASS-BALANCED SUPERVISION

Class imbalance is a common challenge in urban scene understanding, where a few dominant categories significantly outnumber the rest. For instance, most points in urban environments belong to *road*, *building*, and *vegetation*, whereas the point count of other classes is substantially lower. This imbalance limits the model's ability to learn discriminative features across all classes, as the network supervisory loss is overly exposed to only a few major categories (T.-Y. Lin *et al.*, 2017).

To address the issue of class imbalance, we employ a weighted cross-entropy loss function, which is defined as follows:

$$L = -\sum_{i=1}^{N} w_k \log p_i^k, \tag{6.2}$$

where N is the total number of points, k is the GT semantic label of the i^{th} point, p_i^k is the predicted probability of the i^{th} point belonging to its GT category that can be obtained from the network softmax layer. w_k is the weight of the class k and is computed based on the inverse of its relative frequency in the dataset, i.e.,

$$w_k = \sqrt[3]{\frac{N_{max}}{N_k}}. \tag{6.3}$$

where N_k is the number of points of the class k, and N_{max} represents the point count of the class with the highest frequency.

6.4. EXPERIMENTS

6.4.1. DATASET

We use AHN (AHN, 2025), a nationwide ALS point cloud data covering the Netherlands. This open-access resource has been extensively employed in both research and industrial applications. Specifically, we use the AHN4 version. The dataset comprises six semantic classes: *ground*, *building*, *water*, *civil structure*, *high-tension*, and *other*, with vegetation currently categorized under the *other* class. Standard point cloud attributes, such as intensity, are included in the dataset. Although color information is not present in the raw AHN data, it has been post-processed and enriched using publicly available aerial imagery [2].

Due to hardware limitations with processing large-scale point cloud data, the dataset is partitioned into smaller tiles, each measuring 0.25 × 0.3125 km, with a 10-meter overlap between adjacent tiles to mitigate edge effects. For the experiments, 52 tiles are used for training and 8 tiles for testing. Table 6.1 summarizes the distribution of point counts per class in both the training and testing sets, where we have observed a significant class imbalance.

	Total	ground	building	water	civil.	high-tension	other
Train	402.611M	215.135M	66.535M	16.900M	1.126M	0.048M	102.866M
Test	71.071M	37.245M	5.804M	10.382M	0.034M	0.001M	17.606M

Table 6.1: Distribution of point counts across all the classes in the training and testing sets.

6.4.2. IMPLEMENTATION DETAILS AND HYPERPARAMETERS

We adopt KP-Conv (Thomas *et al.*, 2019) as the backbone network for point-wise feature learning. KP-Conv uses a voxel subsampling strategy to reduce the number of input points. We set the voxelization grid size as 20cm (Q. Hu *et al.*, 2021), and the number of kernels as 15. The network is trained for 300 epochs using a batch size of 6, with the initial learning rate set to 0.01.

Table 6.2 presents the hyperparameters we use in the experiments. For the confidence measuring (Section 6.3.1), most hyperparameter values are determined based on geometric priors commonly observed in urban environments. For instance, the spherical neighborhood radius is set to $r = 0.5m$, a value that encompasses the

[2]The colorized AHN point cloud can be accessed at: https://geotiles.citg.tudelft.nl/

spatial extent of urban objects while ensuring a sufficient number of neighboring points for local analysis. The height threshold $\epsilon_h = 2.0m$ and NDVI threshold $\epsilon_v = 0.3$ are selected to effectively distinguish buildings from other urban structures such as vegetation. For the online learning (Section 6.3.2), we set high confidence thresholds to ensure that only the most reliable point samples are used for model updates. Additionally, we set the warm-up period to 150 epochs to allow the network to reach a partially converged state before introducing the online learning mechanism.

Confidence measurement (Setion 6.3.1)				Online learning (Section 6.3.2)			
r	N_{min}	ϵ_h	ϵ_v	c_1	c_2	e_1	e_2
0.5	5	2.0	0.3	0.9	0.99	150	300

Table 6.2: Details of hyperparameters.

6.4.3. QUANTITATIVE RESULTS

To examine the effectiveness of the proposed online learning mechanism for refining point cloud segmentation, we compare its performance against the baseline KP-Conv backbone network (Thomas *et al.*, 2019). Both models are trained using a class-balanced loss function, as detailed in Section 6.3.3. To ensure a fair comparison, identical hyperparameters and training configurations are applied to both networks. We evaluate the segmentation performance using standard metrics, including OA, mIoU, and per-category IoU scores, as described in Section 3.4.3. Due to the high variations in network training, each experimental setup is repeated three times, and the average scores are reported.

Method	OA(%)	**mIoU(%)**	other	ground	building	water	high.	civil.
Baseline	94.8	63.8	**86.6**	94.8	73.5	98.1	27.4	2.6
+ Online	**95.1**	**65.0**	85.4	94.8	**75.4**	**98.4**	**30.4**	**5.7**

Table 6.3: Segmentation results achieved using height and intensity as input features. OA (%), mIoU (%), and per-category IoU scores are reported. We average scores over three training runs to account for network performance variations.

Table 6.3 presents the segmentation performance of the baseline network and its counterpart network augmented with the online learning mechanism, both utilizing height and intensity as input features. The enhanced model demonstrates improved performance over the baseline, achieving gains of 0.3% in OA and 1.2% in mIoU. Among the six categories, five exhibit superior or comparable performance, revealing that our proposed method effectively facilitates network feature learning by prioritizing high-confidence samples during training. In particular, substantial performance improvements are observed in two minority classes, *high-tension* and

civil structure. This indicates the method's ability to handle underrepresented classes. However, for the *other* category, we observe a performance decline of 1.2% mIoU. This is likely attributed to the fact that the *other* class is dominated by vegetation points, which tend to receive lower confidence scores due to their sparse and irregular spatial distribution. As a result, fewer point samples from *other* are incorporated during training, leading to a decreased segmentation performance.

As described in Section 6.4.1, the raw AHN data have been post-processed and enriched with color information by aligning the original point clouds with publicly available aerial imagery. To further analyze the impact of this supplementary color information on the network's performance, we incorporate color channels as additional input features. Table 6.4 reports the performance results for both the baseline network and the network enhanced with the proposed online learning mechanism, trained using the combined height, intensity, and color features.

Method	OA(%)	**mIoU(%)**	other	ground	building	water	high.	civil.
Baseline	**94.8**	**66.9**↑	**87.2**↑	**94.5**↓	**75.8**↑	**96.5**↓	**44.5**↑	**2.7**↑
+ Online	93.9↓	61.5↓	85.3↓	94.2↓	66.0↓	95.4↓	25.6↓	2.4↓

Table 6.4: Segmentation results achieved using height, intensity, and supplementary color information as input features. OA (%), mIoU (%), and per-category IoU scores are reported. We average scores over three training runs to account for network performance variations. We use ↑ to indicate performance improvements compared to the same model trained using only height and intensity features (Table 6.3), and ↓ for performance degradations.

When incorporating supplementary color information as the network input features, our online-learning enhanced network performs worse than the baseline across all metrics. Moreover, while the addition of color features leads to an overall improvement in the baseline network's performance compared to using only height and intensity, it results in a significant performance degradation for the online learning–enhanced network among all six categories.

This observation illustrates one of the significant limitations in our proposed online learning mechanism: Our method strongly depends on the quality of input features. As the color information is derived from aerial imagery, it may contain artifacts resulting from factors such as occlusion or misalignment. The online learning approach is particularly sensitive to such artifacts and imperfections, as it selectively utilizes only a subset of high-confidence points for training, rather than leveraging the entire point cloud. Consequently, these wrong visual cues from supplementary colors can significantly influence the learning process and degrade overall performance.

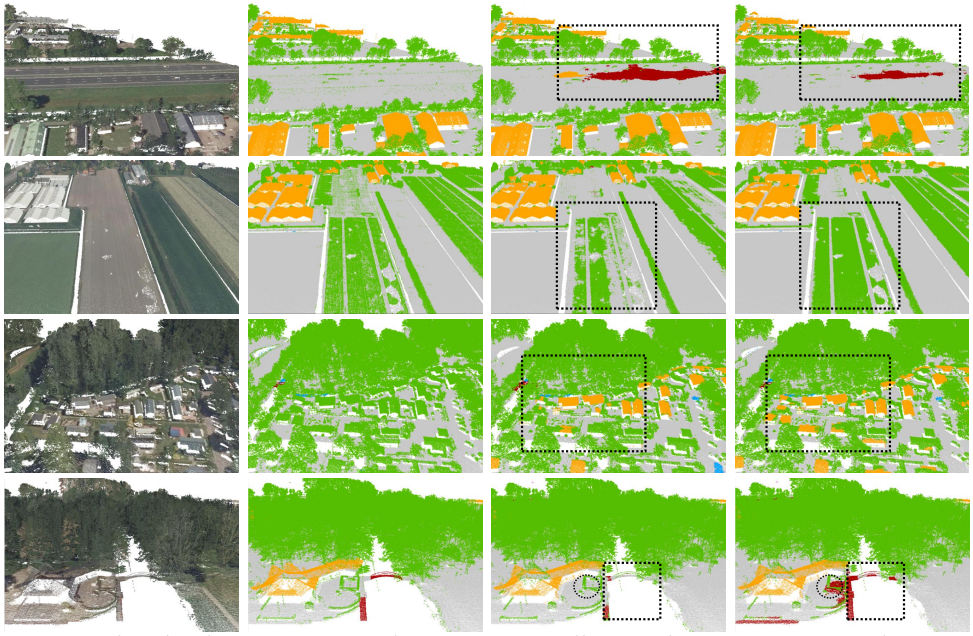


Figure 6.4: Qualitative results achieved on the AHN dataset (AHN, 2025), using height and intensity as input features. Since AHN is a real-world dataset, its annotations may contain noise and errors. Our improvements are highlighted with black-dotted boxes. Minor segmentation noise in some local regions is marked by black-dotted circles.

6.4.4. QUALITATIVE RESULTS

Figure 6.4 presents our qualitative results achieved on the AHN dataset, with the model trained using only the height and intensity features. Figure 6.5 presents qualitative results achieved using height, intensity, and supplementary color information as input features. We use the same color schemes to render the scenes in both figures: ■ for *building*, ■ for *ground*, ■ for *high-tension*, ■ for *water*, ■ for *civil structure*, and ■ for *other*.

When trained with height and intensity features (Figure 6.4), our method consistently outperforms the baseline by reducing segmentation errors. We can also detect urban objects, such as grasslands and bridges, with improved geometric completeness. This performance gain is attributed to our proposed online learning mechanism, which selects only the high-confidence samples to participate in network training, enhancing the robustness and discriminative capacity of the learned feature representations. It is noteworthy that real-world datasets often contain annotation inaccuracies. For example, in the third row of Figure 6.4, all building points are incorrectly labeled as *other*. In this scene, our method successfully classifies a greater

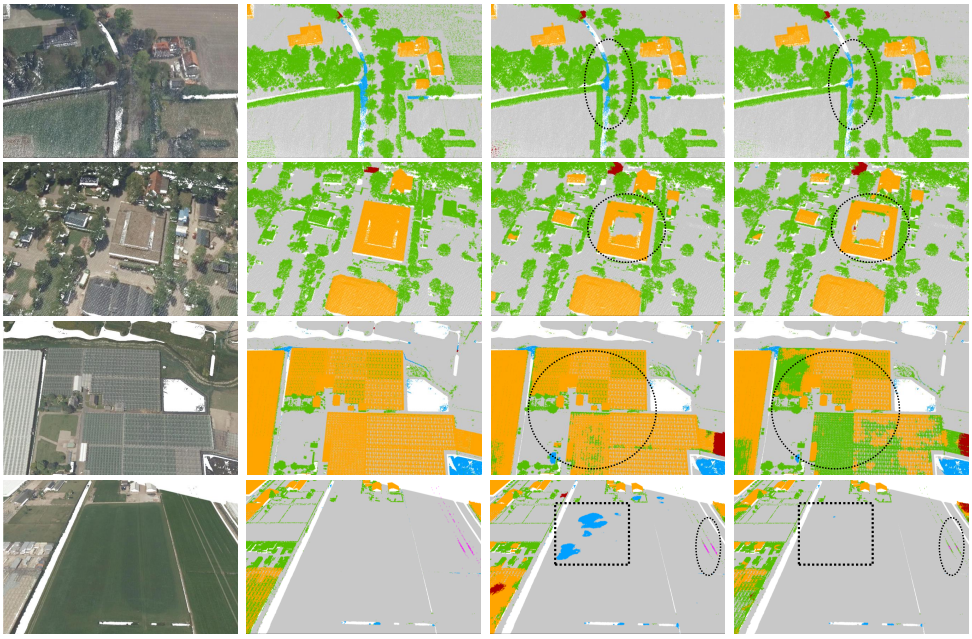


Figure 6.5: Qualitative results achieved on the AHN dataset (AHN, 2025), using height, intensity, and supplementary color as input features. Black-dotted circles highlight our segmentation deficiencies, and black-dotted boxes show regions of improvement.

number of building points, showing its potential to correct misclassification errors and address annotation artifacts in practical applications.

When trained using height, intensity, and supplementary color features (Figure 6.5), our method performs inferior to the baseline. This is due to the low-quality color features, which often arise from occlusion and misalignment between point clouds and aerial imagery. Our approach is more sensitive to low-quality features, since it takes only a subset of points in network training. For example, in the top row, water points are partially occluded by trees, leading to misclassification of *water* as *ground*. In the third row, our method failed to correctly classify greenhouse structures, as the color characteristics misled the network into assigning the *other* label instead of *building*. However, in certain scenes (row 4), our method can generate smoother and more coherent segmentation results.

6.4.5. LABEL REFINEMENT ON TRAINING DATA

Given good input features, our proposed online learning mechanism enhances the robustness and generalizability of the network by dynamically selecting

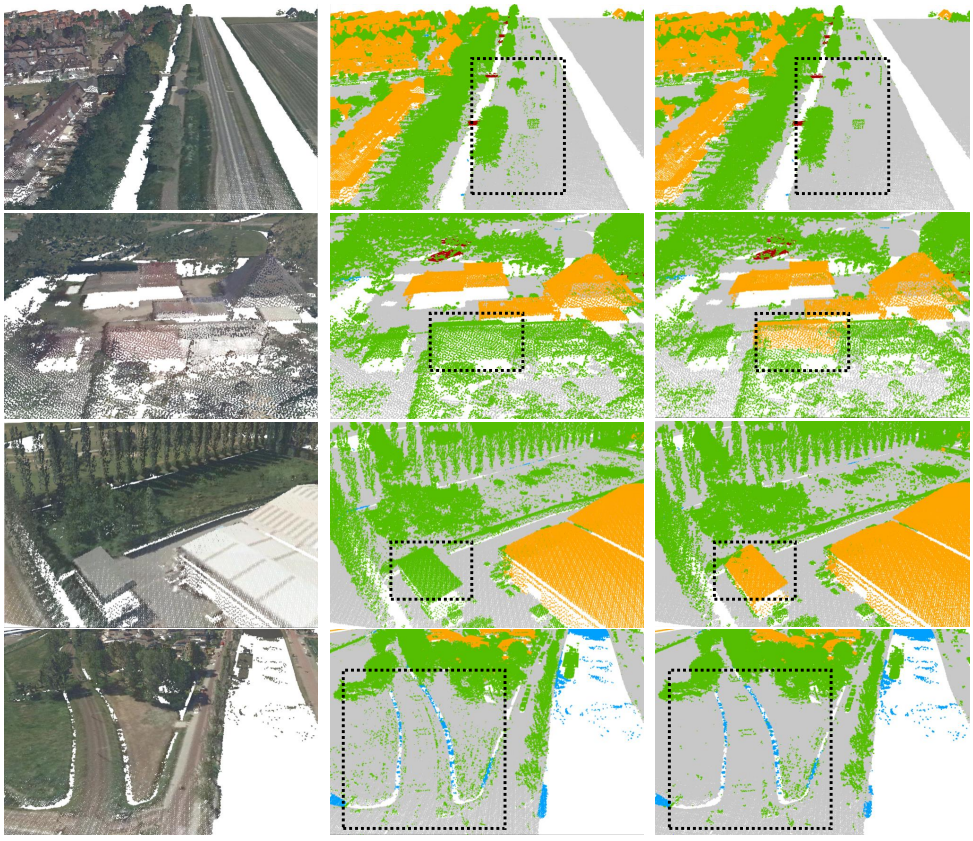


Figure 6.6: Label updates and refinements of the training data with the online learning mechanism.

high-confidence point samples during training. Meanwhile, a natural byproduct of this approach is a cleaned training dataset with annotations progressively refined by pseudo-labels. As detailed in Algorithm 2, when the network makes high-confidence predictions on the set X_U, these new predictions serve as pseudo-labels and gradually replace the original labels. Through this iterative process of updating both the network parameters and the training labels, our online learning framework effectively mitigates label noise, reduces the influence of outliers, and corrects annotation inconsistencies within the dataset.

Figure 6.6 presents representative examples of label refinements in the training set. In the raw data, there are two major types of annotation errors. First, *ground* points are often annotated inconsistently and exhibit label noise. Second, many *building* points are erroneously labeled as *other*, as also demonstrated in row 3 of Figure 6.4. Our online learning framework can effectively address these issues by mitigating

the noise and outliers in the *ground* points and correcting misannotations in the *building* category, which helps us to improve the overall quality of the dataset.

6.4.6. POINT DENSITY ANALYSIS

To compute point-wise confidence scores based on local semantic consistency (Section 6.3.1, Equation 6.1), we set $N_{min} = 5$ to ensure a minimum neighborhood density. In this section, we provide point cloud density analysis to empirically validate our hyperparameter choice.

Figure 6.7 shows the histogram of the point count distribution within a spherical neighborhood of 0.5m, obtained from the training set. The neighboring point counts vary significantly. While most neighborhoods contain between 15 and 30 points, a substantial proportion still falls within the 1–10 point range, indicating that relatively sparse local regions are common in the dataset. However, neighborhoods with fewer than 5 points are rare in the dataset and more susceptible to noise, making them less reliable for estimating local semantic consistency. Setting the neighborhood size threshold too high would exclude many valid points and prevent us from gaining informative local context. On the other hand, setting the threshold too low can lead to overestimation of confidence scores for sparse, potentially noisy regions. Based on this analysis, choosing a threshold of 5 points allows us to achieve a balanced choice, which can effectively filter out extremely sparse and potentially unreliable neighborhoods, while preserving sufficient coverage to capture meaningful local spatial structures.

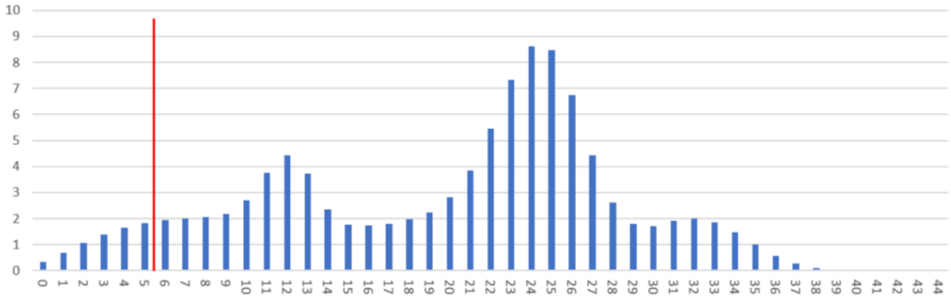


Figure 6.7: Histogram of the distribution of neighborhood point counts. The y-axis represents the number of points, expressed in millions.

6.4.7. BUILDING FOOTPRINT COMPARISON

Besides measuring local semantic consistency, we further incorporate urban building footprint information to enhance the computation of confidence scores. To qualitatively evaluate the fidelity of our extracted building footprints, we overlay them with publicly available 2D BAG building polygons, as shown in Figure 6.8.

(a) Overlaying our building footprints with 2D BAG

(b) Our footprints inferior to 2D BAG (c) Our footprints better than 2D BAG

Figure 6.8: Qualitative comparison between our extracted building footprints and 2D BAG polygons. We use the following colors to render the objects: ▨ for 2D BAG polygons, ■ for our extracted building footprints, and ■ for extracted vegetation footprints.

Overall, there is a strong match between our extracted footprints and the 2D BAG polygons, indicating the effectiveness of the proposed footprint extraction method. In some areas, 2D BAG polygons provide more reliable delineations (Figure 6.8b), particularly in areas where dense tree clusters are wrongly classified as buildings

in our results. This misclassification can be attributed to the fact that simple NDVI filtering might not always distinguish between vegetation and built structures, especially for semi-transparent structures such as greenhouses. Nevertheless, in certain regions, as illustrated in Figure 6.8c, our extracted footprints demonstrate higher accuracy, particularly where the annotation errors in AHN point clouds may affect footprint extraction in the 2D BAG dataset.

6.4.8. LIMITATIONS AND APPLICATIONS

Our proposed online learning framework demonstrates strong effectiveness in addressing real-world point cloud segmentation tasks, exhibiting high potential for refining raw data labels that may be affected by noise, outliers, and annotation errors. However, it suffers from several limitations.

First, this approach is highly sensitive to the quality of input features, as it allows only a subset of the data to participate in network training rather than using the entire dataset. As shown in Table 6.3 and Table 6.4, our method outperforms the baseline when trained with reliable input features such as height and intensity. However, its performance decreases when supplementary color features, potentially introducing wrong visual cues, are incorporated into training. Another limitation lies in the estimation of point-wise confidence scores. We only leverage the geospatial priors from the extracted building footprints to enhance confidence estimation for building-class points. Extending this strategy to incorporate prior knowledge for other object classes could lead to more robust confidence estimation and overall performance improvements. Finally, due to real-world datasets with inherently noisy or incomplete annotations, our evaluation results may lack accuracy, as the available annotation labels cannot be assumed to represent definitive ground truth.

The proposed online learning method is directly applicable to real-world point cloud segmentation tasks, such as the Dutch datasets. It can be used to denoise data, correct prominent annotation errors, and enhance overall data quality. Furthermore, this method can be integrated into semi-automated annotation pipelines to facilitate the annotation and iterative refinement of updated versions of point cloud datasets.

6.5. CONCLUSIONS

In this chapter, we have proposed a confidence-aware online learning framework to explicitly address the challenges of real-world point cloud understanding, specifically accounting for label noise, outliers, and annotation errors. Our framework integrates local semantic consistency measures and geospatial priors associated with specific urban object categories to assess the reliability of existing annotations by assigning a confidence score to each point. These confidence scores are then utilized to guide the online learning process, where the network prioritizes high-confidence samples for training and iteratively refines the annotations of low-confidence points. The proposed approach yields a robustly trained segmentation model and a cleaned

point cloud dataset with improved annotation quality. Extensive experiments have validated the applicability of our approach to Dutch point cloud datasets. However, its performance remains sensitive to the quality of the input features.

Currently, we have evaluated the effectiveness of our proposed approach on the AHN4 point cloud dataset. Given that the most recent release, AHN5, shows comparable characteristics to AHN4 in terms of data acquisition method, spatial distribution, and land cover composition, we expect that our approach will demonstrate similarly robust performance when applied to AHN5 point clouds.

We recommend that future improvements focus on enhancing confidence estimation by integrating more enriched prior knowledge. Meanwhile, leveraging recent advances in computer vision, such as the SAM model (Kirillov *et al.*, 2023), offers a promising direction for extracting high-fidelity object footprints directly from imagery, thereby improving geospatial priors. Another potential direction involves incorporating synthetic point cloud data to mitigate performance limitations associated with minority classes. For instance, artificially augmenting the dataset with samples of underrepresented object categories, such as high-tension lines and civil structures, the network could be trained on a more balanced distribution, improving its ability to learn robust and generalizable features.

7

CONCLUSIONS

This chapter brings together the contributions and key findings of the thesis, which investigates the semantic understanding of urban environments using LiDAR point clouds. The research addresses key challenges in 3D semantic segmentation by examining both local boundary delineation and global contextual representation. It also explores fine-grained instance segmentation of large-scale urban trees. Furthermore, it proposes a data-efficient semantic segmentation technique tailored to real-world datasets. The research questions outlined in Chapter 1 are systematically answered. Finally, potential future research directions are discussed, with the aim of advancing the field and supporting continued progress in semantic urban scene understanding.

7.1. CONTRIBUTIONS AND KEY FINDINGS

Understanding 3D urban scenes is fundamental for various applications such as urban reconstruction, environmental modeling, geolocation, navigation, and autonomous driving. Despite its importance, progress in accurately interpreting 3D urban scenes from point clouds is relatively slow compared to advancements in 2D image-based analysis. To bridge this research gap, this thesis has explored various approaches and tools to 3D point cloud segmentation, which include locally boundary-refined semantic segmentation, globally prototype-enhanced semantic segmentation, fine-grained instance segmentation of specific object categories (e.g., trees), and data-efficient semantic segmentation on real-world datasets that exhibit quality challenges. In this section, I will elaborate on the core contributions, key findings, and answers regarding the proposed research questions.

In **Chapter 3: Local boundary-guided semantic segmentation**, an automatic 3D semantic segmentation algorithm was developed that explicitly mitigates segmentation inconsistencies and ambiguities near urban object boundaries. With the results of this chapter, the research questions proposed in Section 1.3 can now be answered:

1. *What are the underlying reasons of suboptimal boundary delineation in existing point cloud learning approaches?*

   Motivated by the success of deep learning, many learning-based approaches for point cloud semantic segmentation have been introduced (Y. Li *et al.*, 2018; Qi, Yi, *et al.*, 2017; Qian *et al.*, 2022; Thomas *et al.*, 2019). While these methods have demonstrated promising overall performance, they often suffer from suboptimal segmentation accuracy near object boundaries (Section 3.1, Figure 3.1). This limitation is primarily attributed to the loss of fine-grained local information during the network's message passing process. Most concurrent networks adopt an encoding-decoding architectural design, where encoding layers extract hierarchical semantic features and decoding layers propagate these features back to the original spatial resolution. Such a design can result in coarse feature maps, making the network lose object boundary details and fail to generate accurate predictions.

2. *How can boundary priors be effectively integrated to reduce segmentation errors and enhance local-level consistency?*

   I have proposed a boundary-guided semantic segmentation method, where we leverage boundary priors to guide feature propagation during the network decoding to achieve more accurate segmentation in local regions (Section 3.3). This approach involves a multi-task learning framework that jointly performs boundary localization, directional prediction towards object interiors, and semantic segmentation, all sharing a unified feature encoder. Furthermore, to cope with the network information loss, I have designed a lightweight guiding mechanism that fuses the boundary and direction priors to refine the segmentation. By doing so, the proposed method encourages

feature propagation along desired directions and effectively enhances feature representation in local boundary regions. This method can seamlessly integrate into existing segmentation architectures to provide boundary-aware enhancements. Experiments have shown that the proposed approach yields consistent improvements by reducing boundary errors across indoor and outdoor urban scene datasets (Section 3.4).

In **Chapter 4: Global prototype expansion for semantic segmentation**, a novel network module was designed to perform efficient global analysis for point cloud semantic segmentation and urban scene interpretation tasks. With the findings of this chapter, the corresponding research questions can now be answered:

1. *What factors limit the capacity of deep neural networks to perform effective global-level analysis on point clouds?*

   Through a systematic investigation of the architectural paradigms in point cloud learning networks, which include MLP-based (H. Lin *et al.*, 2023; Qi, Su, *et al.*, 2017; Qi, Yi, *et al.*, 2017; Qian *et al.*, 2022), convolution-based (Y. Li *et al.*, 2018; Thomas *et al.*, 2019; M. Xu, Ding, *et al.*, 2021), and transformer-based (Lai *et al.*, 2022; X. Wu *et al.*, 2024; H. Zhao *et al.*, 2021) networks, a common limitation has been observed: the restricted receptive fields. This limitation arises as the network feature operators are often designed to perform locally, meaning each point interacts only with a limited set of neighboring points during feature encoding and aggregation. Therefore, these networks show limited capacity for capturing long-range dependencies and global contextual information, as illustrated in Section 4.2. While incorporating global operators is theoretically feasible, it often incurs significant computational and memory costs, which pose severe scalability challenges.

2. *How can we design a feature operating module that leverages global contextual knowledge to enhance semantic segmentation, while minimizing computational and memory costs?*

   The proposed approach design is fundamentally inspired by the prototype concept, which dates back to cognitive science (Rosch, 1973) and posits that a class can be effectively characterized by its most representative samples. Therefore, prototypes can naturally capture global contextual information. I have designed a novel network module, namely *Prototype Expansion*, to enable global-range feature interactions with negligible increase of network complexity and computational costs (Section 4.3). In this module, I use a single prototype embedding to abstract the characteristics of an object class, which is constructed from training samples and dynamically updated during training. Then, an attention-like operation is performed to model the dependencies between points and global class prototypes in the feature space, according to which the prototypes are projected back to enhance individual point feature representation. Through such a process, PE becomes a global network operator. Meanwhile, it remains computationally efficient,

as each point interacts with only a limited set of prototype embeddings. The experiments have shown that the proposed PE module results in a minimal increase in model complexity and achieves similar inference speed compared to baseline methods. Nevertheless, the PE module consistently yields substantial improvements in segmentation accuracy across various prominent point cloud learning networks on indoor and outdoor scenes, highlighting its effectiveness as a general enhancement for 3D semantic segmentation tasks 4.4.

In **Chapter 5: Structure-aware tree instance segmentation**, a robust deep learning framework was developed for automated 3D instance segmentation of trees in large-scale urban and nature forestry scenes. With the findings of this chapter, the relevant research questions defined in Section 1.3 can now be answered:

1. *Which shape characteristics distinguish trees from other types of urban objects and can be exploited for instance segmentation tasks?*

   Trees present significant challenges for segmentation due to their complex and irregular morphological structures. However, despite their variations in species, size, and shape, all trees share a fundamental structural composition: a stem (or trunk) and a crown. This structural characteristic differentiates trees from other urban objects, enabling instance-level tree segmentation by leveraging key features such as tree stems. Nevertheless, most existing methods for tree instance segmentation, either heuristic-based approaches (Q. Chen *et al.*, 2006; Hakula *et al.*, 2023; Tao *et al.*, 2015; J. Wang *et al.*, 2018) or deep learning-based approaches (Chang *et al.*, 2022; Henrich *et al.*, 2024; T. Jiang, Wang, *et al.*, 2023; P. Wang *et al.*, 2023), fail to explicitly incorporate these structural priors into their segmentation pipelines, as detailed in Section 5.2. Therefore, these methods often suffer from over-segmentation or under-segmentation, especially in densely vegetated and complex urban environments. Recent studies Ning *et al.* (2023) and Pu *et al.* (2023) have explored the localization of tree stems as a strategy for delineating individual trees. They often require additional network learning modules for stem detection, thus resulting in increased model complexity and training costs. Hence, there is a pressing need for a unified framework that integrates tree structural priors directly into the instance segmentation process, enabling more accurate and efficient identification of individual trees.

2. *How to effectively address tree overlap, occlusion, and geometric variations in challenging urban forestry areas?*

   I have developed a novel structure-aware deep learning-based framework for tree instance segmentation in large-scale and complex forestry environments, as demonstrated in Section 5.3. The core idea is to explicitly leverage tree structure priors (e.g., detected tree stems) to enhance the robustness and accuracy of instance segmentation. This proposed framework simultaneously performs (i) semantic segmentation to classify a point as *crown*, *stem*, or *other*; (ii) heatmap prediction to assign a heat value to each point based on

2D Gaussian kernels centered at tree stem locations; (iii) offset prediction to estimate point-wise offset vectors pointing to the instance centroid. These multi-task outputs are then fused to achieve precise localization of tree stems. Last, I use a graph-based shortest path approach to isolate individual tree instances, which not only considers the Euclidean proximity of points but also integrates the predicted directional information. Such a strategy ensures that segmented tree boundaries adhere to the underlying tree structures, even in dense forestry scenes where crowns heavily overlap. Extensive experiments on natural and urban scenes demonstrate that the proposed approach consistently outperforms state-of-the-art techniques (Section 5.4). This significantly reduces over-segmentation and under-segmentation errors, highlighting its effectiveness in addressing the challenges of tree instance segmentation in complex environments.

In **Chapter 6: Confidence-based online learning for real-world point clouds**, a data-efficient learning strategy was designed for interpreting real-world urban point clouds, accounting for degraded data quality, such as noise, outliers, and annotation defects. With the results of this chapter, the relevant research questions in Section 1.3 can now be answered:

1. *How can data uncertainties be measured and leveraged to improve semantic understanding of point clouds in real-world environments?*

   Real-world point cloud datasets naturally contain data noise, outliers, and annotation errors, which pose significant challenges for accurate classification and segmentation. To address these data uncertainties, a confidence-aware online learning framework has been introduced that dynamically adapts to annotation quality during training. The proposed approach leverages local semantic consistency and geospatial priors extracted from aerial imagery to estimate a confidence score for each point, which reflects the reliability of its associated annotation. These confidence scores are then used to dynamically guide the training process, where high-confidence samples are prioritized for learning and the labels of low-confidence samples are iteratively refined based on model predictions. The proposed mechanism is straightforward yet effective in mitigating the adverse effects of noisy labels, enhancing segmentation accuracy without increased computational complexity (Section 6.3). As a result, this method not only yields a robustly trained segmentation model but also produces a cleaned point cloud dataset with improved annotation quality.

2. *How well does the confidence-based approach perform on real-world airborne point clouds, and what are its potential applications?*

   The proposed confidence-based online learning framework has demonstrated its effectiveness and practical applicability on large-scale real-world point cloud datasets, such as the Dutch AHN dataset (AHN, 2025), as demonstrated in Section 6.4. Trained with high-fidelity input features, such as point elevation and intensity, this method enhances the robustness and generalizability of the

segmentation network and consistently reduces segmentation errors. Moreover, the dynamic label refinement mechanism also effectively mitigates annotation noise and errors in the original dataset, improving label quality. However, the proposed method is sensitive to the quality of input features. When trained with lower-fidelity features, the segmentation accuracy tends to decrease. Beyond segmentation performance improvements, the framework can be used as a data denoising tool to correct labeling noise and prominent annotation errors. It can also be integrated into semi-automated annotation pipelines to facilitate efficient annotation and iterative refinement of annotations of point cloud datasets.

7.2. FUTURE WORK

This section provides detailed discussions on potential future research directions in urban understanding using 3D point clouds, in light of the current limitations identified in this thesis.

- **Incorporating high-level structural priors into deep learning approaches**.

  Deep learning approaches have greatly surpassed classical machine learning methods as they can learn powerful feature representations directly from raw data inputs by representing objects as a hierarchy of concepts. However, achieving good performance using deep learning typically requires large-scale annotated datasets, prolonged training cycles, and the design of complex feature learning architectures. Furthermore, deep learning is often viewed as a black box, where the internal feature learning process is challenging to interpret due to the complex multi-layer structure of neural networks. This lack of transparency makes deep learning models sensitive to data distribution shifts, hyperparameter tuning, and overfitting. Incorporating domain knowledge, such as structural priors or geometrical constraints, into the network learning process would enhance the robustness and interpretability of the learned feature representation and therefore lead to more reliable and generalizable model performance. In Chapter 3, I have incorporated object boundary priors to improve segmentation accuracy. However, boundaries represent low-level geometric cues. Some recent studies have integrated higher-level priors, including the continuous unsigned distance field (Z. Yang *et al.*, 2022) and graph topological structures (Landrieu & Simonovsky, 2018; Robert *et al.*, 2023). Despite these advancements, research in leveraging high-level priors remains limited. Future research has significant potential to investigate and incorporate high-level domain knowledge to enhance the robustness, interpretability, and overall efficacy of deep learning models for understanding 3D point clouds.

- **Combining multi-modality data sources**.

  In this thesis, I solely use LiDAR point clouds as input data for interpreting and analyzing urban environments. Although auxiliary data sources, such as aerial

imagery, are employed in Chapter 6, their use is limited to supporting the evaluation of point-wise confidence scores, rather than being fully integrated into the learning framework. A promising future research direction is developing deep learning methods that effectively fuse 2D imagery and 3D point cloud data, where the two distinct modalities mutually complement each other to enhance the overall accuracy and robustness of urban scene analysis tasks. While deep learning on 3D point clouds remains a challenging and evolving field, deep learning techniques for 2D images have matured rapidly, driven by advances in network architectures (Krizhevsky *et al.*, 2012; Z. Liu *et al.*, 2021) and the availability of large annotated datasets (Deng *et al.*, 2009). Moreover, the recent emergence of large 2D foundation models (Kirillov *et al.*, 2023) offers new opportunities to transfer knowledge from image-based tasks to 3D vision problems, potentially benefiting areas such as semi-supervised and unsupervised 3D learning. Overall, developing a multi-modal deep learning framework that incorporates both 2D and 3D input holds significant potential for advancing urban scene analysis in the future.

- **Joint segmentation of urban tree instances and tree parts**.

  This thesis primarily investigates deep learning-based instance segmentation of large-scale trees. Although it leverages the detection of key structural components, such as tree stems, to facilitate robust localization of individual tree instances, it does not explicitly address the task of tree part segmentation. Compared to instance segmentation, part-level tree segmentation provides a more granular understanding of tree morphology, capturing detailed structural information within each tree. However, this increased level of detail also introduces greater segmentation challenges. Predominant methods typically take a single tree as input and subsequently perform tree-part decomposition (Y. Liu *et al.*, 2021; Shen, 2024). Nevertheless, significant research potential exists in jointly addressing tree instance segmentation and tree part segmentation within a unified learning framework. Since tree parts are inherently nested within tree instances, a joint formulation allows the two tasks to complement and reinforce each other. As demonstrated in Chapter 5, successfully identifying tree parts can provide valuable cues to enhance the accuracy of instance segmentation, particularly in complex forestry scenes with dense crown overlap. Future research can explore integrated approaches that simultaneously perform instance-level and part-level segmentation directly from large-scale forestry point clouds. Such a unified framework would enable fine-grained recognition of tree structures while leveraging part-level information to improve instance delineation. This would facilitate a comprehensive and detailed urban forestry analysis, contributing to more precise assessments of forestry resources, biomass estimation, and other critical environmental indicators.

Finally, I would like to end this thesis with a reflection on the broader research context and future perspectives. When I began my Ph.D. in 2020, deep learning on 3D point clouds for urban scene analysis was an emerging research area. Since then, this field has experienced rapid progress driven by advances in 3D deep learning

architectures (Lai *et al.*, 2022; Thomas *et al.*, 2019; X. Wu *et al.*, 2024; H. Zhao *et al.*, 2021). However, the evolution of 3D deep learning remains comparatively slower than that of 2D deep learning. Furthermore, many existing methods encounter challenges in generalizability and scalability when applied to real-world urban environments characterized by large spatial extents and high data uncertainty. I hope that the methodological contributions presented in this thesis will advance the fundamental understanding of 3D point cloud learning, as well as support the development of practical solutions for real-world urban applications.

BIBLIOGRAPHY

Agugiaro, G. (2016). Energy planning tools and citygml-based 3d virtual city models: Experiences from trento (italy). *Applied Geomatics*, *8*(1), 41–56. https://doi.org/10.1007/s12518-015-0163-2

AHN. (2025). Actueel hoogtebestand nederland [Accessed: 2025-01-14]. https://www.ahn.nl/

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., & Maximov, Y. (2025). Self-training: A survey. *Neurocomputing*, *616*, 128904. https://doi.org/10.2139/ssrn.4875054

Ammar, A., Koubaa, A., & Benjdira, B. (2021). Deep-learning-based automated palm tree counting and geolocation in large farms from aerial geotagged images. *Agronomy*, *11*(8), 1458. https://doi.org/10.3390/agronomy11081458

Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016). State of the art of virtual reality technology. *IEEE Aerospace Conference*, 1–19. https://doi.org/10.1109/aero.2016.7500674

Arief, H. A., Strand, G.-H., Tveite, H., & Indahl, U. G. (2018). Land cover segmentation of airborne lidar data using stochastic atrous network. *Remote Sensing*, *10*(6), 973. https://doi.org/10.3390/rs10060973

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., & Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. *IEEE Conference on Computer Vision and Pattern Recognition*, 1534–1543. https://doi.org/10.1109/cvpr.2016.170

Arroyo Ohori, K., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., & Stoter, J. (2017). Towards an integration of gis and bim data: What are the geometric and topological issues? *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*, 1–8. https://doi.org/10.5194/isprs-annals-iv-4-w5-1-2017

Ayrey, E., Fraver, S., Kershaw Jr, J. A., Kenefic, L. S., Hayes, D., Weiskittel, A. R., & Roth, B. E. (2017). Layer stacking: A novel algorithm for individual forest tree segmentation from lidar point clouds. *Canadian Journal of Remote Sensing*, *43*(1), 16–27. https://doi.org/10.1080/07038992.2017.1252907

BAG. (2025). Basisregistratie adressen en gebouwen [Accessed: 2025-07-01]. https://www.kadaster.nl/zakelijk/registraties/basisregistraties/bag

Berg, A., Oskarsson, M., & O'Connor, M. (2022). Points to patches: Enabling the use of self-attention for 3d shape recognition. *International Conference on Pattern Recognition*, 528–534. https://doi.org/10.1109/icpr56361.2022.9956172

Bertasius, G., Shi, J., & Torresani, L. (2016). Semantic segmentation with boundary neural fields. *IEEE Conference on Computer Vision and Pattern Recognition*, 3602–3610. https://doi.org/10.1109/cvpr.2016.392

Beucher, S. (1979). Use of watersheds in contour detection. *International Workshop on Image Processing*, 17–21.

Biehler, M., Sun, Y., Kode, S., Li, J., & Shi, J. (2023). Plural: 3d point cloud transfer learning via contrastive learning with augmentations. *IEEE Transactions on Automation Science and Engineering*, *21*(4), 7550–7561. https://doi.org/10.1109/tase.2023.3345807

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, *4*(4), 2842–2889. https://doi.org/10.3390/ijgi4042842

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint*. https://doi.org/10.48550/arXiv.2004.10934

Bouzas, V., Ledoux, H., & Nan, L. (2020). Structure-aware building mesh polygonization. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*, 432–442. https://doi.org/10.1016/j.isprsjprs.2020.07.010

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6154–6162. https://doi.org/10.1109/cvpr.2018.00644

Cappelle, C., El Najjar, M. E., Charpillet, F., & Pomorski, D. (2012). Virtual 3d city model for navigation in urban areas. *Journal of Intelligent & Robotic Systems*, *66*, 377–399. https://doi.org/10.1007/s10846-011-9594-0

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, *33*, 9912–9924. https://arxiv.org/abs/2006.09882

Chang, L., Fan, H., Zhu, N., & Dong, Z. (2022). A two-stage approach for individual tree segmentation from tls point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 8682–8693. https://doi.org/10.1109/jstars.2022.3212445

Chen, D., Wang, R., & Peethambaran, J. (2017). Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(12), 7032–7052. https://doi.org/10.1109/tgrs.2017.2738439

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. https://doi.org/10.1109/tpami.2017.2699184

Chen, Q., Baldocchi, D., Gong, P., & Kelly, M. (2006). Isolating individual trees in a savanna woodland using small footprint lidar data. *Photogrammetric Engineering & Remote Sensing*, *72*(8), 923–932. https://doi.org/10.14358/pers.72.8.923

Chen, S., Fang, J., Zhang, Q., Liu, W., & Wang, X. (2021). Hierarchical aggregation for 3d instance segmentation. *IEEE/CVF International Conference on Computer Vision*, 15467–15476. https://doi.org/10.1109/iccv48922.2021.01518

Chen, X., Jiang, K., Zhu, Y., Wang, X., & Yun, T. (2021). Individual tree crown segmentation directly from uav-borne lidar data using the pointnet of deep learning. *Forests*, *12*(2), 131. https://doi.org/10.3390/f12020131

Cheng, B., Schwing, A., & Kirillov, A. (2021). Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, *34*, 17864–17875. https://arxiv.org/abs/2107.06278

Choy, C., Gwak, J., & Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3075–3084. https://doi.org/10.1109/cvpr.2019.00319

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. *Medical Image Computing and Computer-Assisted Intervention*, 424–432. https://doi.org/10.1007/978-3-319-46723-8_49

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/tit.1967.1053964

Crespo, J., Castillo, J. C., Mozos, O. M., & Barber, R. (2020). Semantic information for robot navigation: A survey. *Applied Sciences*, *10*(2), 497. https://doi.org/10.3390/app10020497

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314. https://doi.org/10.1007/bf02134016

Dabove, P., Daud, M., & Olivotto, L. (2024). Revolutionizing urban mapping: Deep learning and data fusion strategies for accurate building footprint segmentation. *Scientific Reports*, *14*(1), 13510. https://doi.org/10.1038/s41598-024-64231-0

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839. https://doi.org/10.1109/cvpr.2017.261

Dassot, M., Constant, T., & Fournier, M. (2011). The use of terrestrial lidar technology in forest science: Application fields, benefits and challenges. *Annals of Forest Science*, *68*, 959–974. https://doi.org/10.1007/s13595-011-0102-2

DeFries, R. S., & Townshend, J. (1994). Ndvi-derived land cover classifications at a global scale. *International Journal of Remote Sensing*, *15*(17), 3567–3586. https://doi.org/10.1080/01431169408954345

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/cvpr.2009.5206848

Ding, H., Jiang, X., Liu, A. Q., Thalmann, N. M., & Wang, G. (2019). Boundary-aware feature propagation for scene segmentation. *IEEE/CVF International Conference on Computer Vision*, 6819–6829. https://doi.org/10.1109/iccv.2019.00692

Dong, S., Lin, G., & Hung, T.-Y. (2022). Learning regional purity for instance segmentation on 3d point clouds. *European Conference on Computer Vision*, 56–72. https://doi.org/10.1007/978-3-031-20056-4_4

Dosovitskiy, A. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*. https://arxiv.org/abs/2010.11929

Dralle, K., & Rudemo, M. (1996). Stem number estimation by kernel smoothing of aerial photos. *Canadian Journal of Forest Research*, *26*(7), 1228–1236. https://doi.org/10.1139/x26-137

Du, S., Lindenbergh, R., Ledoux, H., Stoter, J., & Nan, L. (2019). Adtree: Accurate, detailed, and automatic modelling of laser-scanned trees. *Remote Sensing*, *11*(18), 2074. https://doi.org/10.3390/rs11182074

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. *IEEE/CVF International Conference on Computer Vision*, 6569–6578. https://doi.org/10.1109/iccv.2019.00667

Fan, G., Nan, L., Dong, Y., Su, X., & Chen, F. (2020). Adqsm: A new method for estimating above-ground biomass from tls point clouds. *Remote Sensing*, *12*(18), 3089. https://doi.org/10.3390/rs12183089

Fu, C., Li, Q., Xu, K., & Wu, J. (2023). Point cloud analysis for ml-based malicious traffic detection: Reducing majorities of false positive alarms. *ACM SIGSAC Conference on Computer and Communications Security*, 1005–1019. https://doi.org/10.1145/3576915.3616631

García-Sánchez, C., van Beeck, J., & Gorlé, C. (2018). Predictive large eddy simulations for urban flows: Challenges and opportunities. *Building and Environment*, *139*, 146–156. https://doi.org/10.1016/j.buildenv.2018.05.007

Geiger, A., Lauer, M., & Urtasun, R. (2011). A generative model for 3d urban scene understanding from movable platforms. *IEEE Conference on Computer Vision and Pattern Recognition*, 1945–1952. https://doi.org/10.1109/cvpr.2011.5995641

Gong, J., Xu, J., Tan, X., Zhou, J., Qu, Y., Xie, Y., & Ma, L. (2021). Boundary-aware geometric encoding for semantic segmentation of point clouds. *AAAI Conference on Artificial Intelligence*, *35*(2), 1424–1432. https://doi.org/10.1609/aaai.v35i2.16232

Goodfellow, I. (2016). *Deep learning*. MIT press.

Graham, B., Engelcke, M., & Van Der Maaten, L. (2018). 3d semantic segmentation with submanifold sparse convolutional networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9224–9232. https://doi.org/10.1109/cvpr.2018.00961

Guo, H., Wang, J., Gao, Y., Li, J., & Lu, H. (2016). Multi-view 3d object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, *25*(12), 5526–5537. https://doi.org/10.1109/tip.2016.2609814

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., & Hu, S.-M. (2021). Pct: Point cloud transformer. *Computational Visual Media*, *7*, 187–199. https://doi.org/10.1007/s41095-021-0229-5

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(12), 4338–4364.

Gupta, S., Weinacker, H., & Koch, B. (2010). Comparative analysis of clustering-based approaches for 3-d single tree detection using airborne fullwave lidar data. *Remote Sensing*, *2*(4), 968–989. https://doi.org/10.3390/rs2040968

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., & 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. https://doi.org/10.5194/isprs-annals-iv-1-w1-91-2017

Hakula, A., Ruoppa, L., Lehtomäki, M., Yu, X., Kukko, A., Kaartinen, H., Taher, J., Matikainen, L., Hyyppä, E., Luoma, V., et al. (2023). Individual tree segmentation and species classification using high-density close-range multispectral laser scanning data. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, *9*, 100039. https://doi.org/10.1016/j.ophoto.2023.100039

Hamraz, H., Jacobs, N. B., Contreras, M. A., & Clark, C. H. (2019). Deep learning for conifer/deciduous classification of airborne lidar 3d point clouds representing individual trees. *ISPRS Journal of Photogrammetry and Remote Sensing*, *158*, 219–230. https://doi.org/10.1016/j.isprsjprs.2019.10.011

Hayder, Z., He, X., & Salzmann, M. (2017). Boundary-aware instance segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 5696–5704. https://doi.org/10.1109/cvpr.2017.70

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *IEEE International Conference on Computer Vision*, 2961–2969. https://doi.org/10.1109/iccv.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. https://doi.org/10.1109/cvpr.2016.90

Heinzel, J., & Huber, M. O. (2018). Constrained spectral clustering of individual trees in dense forest using terrestrial laser scanning data. *Remote Sensing*, *10*(7), 1056. https://doi.org/10.3390/rs10071056

Henrich, J., van Delden, J., Seidel, D., Kneib, T., & Ecker, A. S. (2024). Treelearn: A deep learning method for segmenting individual trees from ground-based lidar forest point clouds. *Ecological Informatics*, *84*. https://doi.org/10.1016/j.ecoinf.2024.102888

Hou, J., Dai, A., & Nießner, M. (2019). 3d-sis: 3d semantic instance segmentation of rgb-d scans. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4421–4430. https://doi.org/10.1109/cvpr.2019.00455

Hou, J., Graham, B., Nießner, M., & Xie, S. (2021). Exploring data-efficient 3d scene understanding with contrastive scene contexts. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15587–15597. https://doi.org/10.1109/cvpr46437.2021.01533

Hsu, L.-T., Gu, Y., & Kamijo, S. (2015). Nlos correction/exclusion for gnss measurement using raim and city building models. *Sensors*, *15*(7), 17329–17349. https://doi.org/10.3390/s150717329

Hu, Q., Yang, B., Fang, G., Guo, Y., Leonardis, A., Trigoni, N., & Markham, A. (2022). Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. *European Conference on Computer Vision*, 600–619. https://doi.org/10.1007/978-3-031-19812-0_35

Hu, Q., Yang, B., Khalid, S., Xiao, W., Trigoni, N., & Markham, A. (2021). Towards semantic segmentation of urban-scale 3d point clouds: A dataset, benchmarks and challenges. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4977–4987. https://doi.org/10.1109/cvpr46437.2021.00494

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., & Markham, A. (2020). Randla-net: Efficient semantic segmentation of large-scale point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117. https://doi.org/10.1109/cvpr42600.2020.01112

Hu, Z., Zhen, M., Bai, X., Fu, H., & Tai, C.-l. (2020). Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. *European Conference on Computer Vision*, 222–239. https://doi.org/10.1007/978-3-030-58565-5_14

Huang, J., Stoter, J., Peters, R., & Nan, L. (2022). City3d: Large-scale building reconstruction from airborne lidar point clouds. *Remote Sensing*, *14*(9), 2254. https://doi.org/10.3390/rs14092254

Huang, Z., Huang, L., Gong, Y., Huang, C., & Wang, X. (2019). Mask scoring r-cnn. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6409–6418. https://doi.org/10.1109/cvpr.2019.00657

Hyyppä, J., Kelle, O., Lehikoinen, M., & Inkinen, M. (2001). A segmentation-based method to retrieve stem volume estimates from 3-d tree height models produced by laser scanners. *IEEE Transactions on Geoscience and Remote Sensing*, *39*(5), 969–975. https://doi.org/10.1109/36.921414

Hyyppä, J., Yu, X., Hyyppä, H., Vastaranta, M., Holopainen, M., Kukko, A., Kaartinen, H., Jaakkola, A., Vaaja, M., Koskinen, J., et al. (2012). Advances in forest inventory using airborne laser scanning. *Remote Sensing*, *4*(5), 1190–1207. https://doi.org/10.3390/rs4051190

James, M. R., & Robson, S. (2012). Straightforward reconstruction of 3d surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, *117*(F3). https://doi.org/10.1029/2011jf002289

Jiang, L., Zhao, H., Liu, S., Shen, X., Fu, C.-W., & Jia, J. (2019). Hierarchical point-edge interaction network for point cloud semantic segmentation. *IEEE/CVF International Conference on Computer Vision*, 10433–10441. https://doi.org/10.1109/iccv.2019.01053

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., & Jia, J. (2020). Pointgroup: Dual-set point grouping for 3d instance segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4867–4876. https://doi.org/10.1109/cvpr42600.2020.00492

Jiang, T., Liu, S., Zhang, Q., Xu, X., Sun, J., & Wang, Y. (2023). Segmentation of individual trees in urban mls point clouds using a deep learning framework based on cylindrical convolution network. *International Journal of Applied Earth Observation and Geoinformation*, *123*, 103473. https://doi.org/10.1016/j.jag.2023.103473

Jiang, T., Wang, Y., Liu, S., Zhang, Q., Zhao, L., & Sun, J. (2023). Instance recognition of street trees from urban point clouds using a three-stage neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, *199*, 305–334. https://doi.org/10.1016/j.isprsjprs.2023.04.010

Kadaster. (2025). Kadaster: The netherlands' cadastre, land registry and mapping agency [Accessed: 2025-02-25]. https://www.kadaster.nl/

Kemeç, S., & Duzgun, H. (2006). Use of 3d visualization in natural disaster risk assessment for urban areas. *Innovations in 3D Geo Information Systems*, 557–566. https://doi.org/10.1007/978-3-540-36998-1_43

Khoshelham, K., Elberink, S. O., & Xu, S. (2013). Segment-based classification of damaged building roofs in aerial laser scanning data. *IEEE Geoscience and Remote Sensing Letters*, *10*(5), 1258–1262. https://doi.org/10.1109/lgrs.2013.2257676

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems*, *33*, 18661–18673. https://arxiv.org/abs/2004.11362

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. *IEEE/CVF International Conference on Computer Vision*, 4015–4026. https://doi.org/10.1109/iccv51070.2023.00371

Kolodiazhnyi, M., Vorontsova, A., Konushin, A., & Rukhovich, D. (2024). Oneformer3d: One transformer for unified point cloud segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20943–20953. https://doi.org/10.1109/cvpr52733.2024.01979

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*. https://doi.org/10.1145/3065386

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., & Jia, J. (2022). Stratified transformer for 3d point cloud segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8500–8509. https://doi.org/10.1109/cvpr52688.2022.00831

Lakoff, G. (2007). Cognitive models and prototype theory. *The Cognitive Linguistics Reader*, 130–167.

Lalonde, J.-F., Unnikrishnan, R., Vandapel, N., & Hebert, M. (2005). Scale selection for classification of point-sampled 3d surfaces. *International Conference on 3-D Digital Imaging and Modeling*, 285–292. https://doi.org/10.1109/3dim.2005.71

Landrieu, L., Raguet, H., Vallet, B., Mallet, C., & Weinmann, M. (2017). A structured regularization framework for spatially smoothing semantic labelings of 3d

point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *132*, 102–118. https://doi.org/10.1016/j.isprsjprs.2017.08.010

Landrieu, L., & Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4558–4567. https://doi.org/10.1109/cvpr.2018.00479

Lateef, F., & Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, *338*, 321–348. https://doi.org/10.1016/j.neucom.2019.02.003

Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., & Felsberg, M. (2017). Deep projective 3d semantic segmentation. *Computer Analysis of Images and Patterns*, 95–107. https://doi.org/10.1007/978-3-319-64689-3_8

Le Toan, T., Quegan, S., Davidson, M., Balzter, H., Paillou, P., Papathanassiou, K., Plummer, S., Rocca, F., Saatchi, S., Shugart, H., et al. (2011). The biomass mission: Mapping global forest biomass to better understand the terrestrial carbon cycle. *Remote Sensing of Environment*, *115*(11), 2850–2860. https://doi.org/10.1016/j.rse.2011.03.020

Ledoux, H., Biljecki, F., Dukai, B., Kumar, K., Peters, R., Stoter, J., & Commandeur, T. (2021). 3dfier: Automatic reconstruction of 3d city models. *Journal of Open Source Software*, *6*(57), 2866. https://doi.org/10.21105/joss.02866

Lee, H., Slatton, K. C., Roth, B. E., & Cropper Jr, W. (2010). Adaptive clustering of airborne lidar data to segment individual tree crowns in managed pine forests. *International Journal of Remote Sensing*, *31*(1), 117–139. https://doi.org/10.1080/01431160902882561

Li, D., Liu, G., & Liao, S. (2015). Solar potential in urban residential buildings. *Solar Energy*, *111*, 225–235. https://doi.org/10.1016/j.solener.2014.10.045

Li, H., Sun, Z., Wu, Y., & Song, Y. (2021). Semi-supervised point cloud segmentation using self-training with label confidence prediction. *Neurocomputing*, *437*, 227–237. https://doi.org/10.1016/j.neucom.2021.01.091

Li, J., & Dong, Q. (2023). Open-set semantic segmentation for point clouds via adversarial prototype framework. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9425–9434. https://doi.org/10.1109/cvpr52729.2023.00909

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. (2018). Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems*, *31*. https://arxiv.org/abs/1801.07791

Lin, H., Zheng, X., Li, L., Chao, F., Wang, S., Wang, Y., Tian, Y., & Ji, R. (2023). Meta architecture for point cloud analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17682–17691. https://doi.org/10.1109/cvpr52729.2023.01696

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *IEEE International Conference on Computer Vision*, 2980–2988. https://doi.org/10.1109/iccv.2017.324

Liu, L., Zhang, L., Ma, J., Zhang, L., Zhang, X., Xiao, Z., & Yang, L. (2010). An improved line-of-sight method for visibility analysis in 3d

complex landscapes. *Science China Information Sciences*, *53*, 2185–2194. https://doi.org/10.1007/s11432-010-4090-x

Liu, M., Zhou, Y., Qi, C. R., Gong, B., Su, H., & Anguelov, D. (2022). Less: Label-efficient semantic segmentation for lidar point clouds. *European Conference on Computer Vision*, 70–89. https://doi.org/10.1007/978-3-031-19842-7_5

Liu, Y., Guo, J., Benes, B., Deussen, O., Zhang, X., & Huang, H. (2021). Treepartnet: Neural decomposition of point clouds for 3d tree reconstruction. *ACM Transactions on Graphics*, *40*(6), 1–16. https://doi.org/10.1145/3478513.3480486

Liu, Y., Kong, L., Cen, J., Chen, R., Zhang, W., Pan, L., Chen, K., & Liu, Z. (2024). Segment any point cloud sequences by distilling vision foundation models. *Advances in Neural Information Processing Systems*, *36*. https://arxiv.org/abs/2306.09347

Liu, Y., Cheng, M.-M., Hu, X., Wang, K., & Bai, X. (2017). Richer convolutional features for edge detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 3000–3009. https://doi.org/10.1109/cvpr.2017.622

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. (2022). Swin transformer v2: Scaling up capacity and resolution. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12009–12019. https://doi.org/10.1109/cvpr52688.2022.01170

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *IEEE/CVF International Conference on Computer Vision*, 10012–10022. https://doi.org/10.1109/iccv48922.2021.00986

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., & El-Sana, J. (2010). Automatic reconstruction of tree skeletal structures from point clouds. *ACM SIGGRAPH Asia*, 1–8. https://doi.org/10.1145/1866158.1866177

Lodha, S. K., Kreps, E. J., Helmbold, D. P., & Fitzpatrick, D. (2006). Aerial lidar data classification using support vector machines (svm). *International Symposium on 3D Data Processing, Visualization, and Transmission*, 567–574. https://doi.org/10.1109/3dpvt.2006.23

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *IEEE International Conference on Computer Vision*, *2*, 1150–1157. https://doi.org/10.1109/iccv.1999.790410

Lu, J., Deng, J., Wang, C., He, J., & Zhang, T. (2023). Query refinement transformer for 3d instance segmentation. *IEEE/CVF International Conference on Computer Vision*, 18516–18526. https://doi.org/10.1109/iccv51070.2023.01697

Luo, H., Khoshelham, K., Chen, C., & He, H. (2021). Individual tree extraction from urban mobile laser scanning point clouds using deep pointwise direction embedding. *ISPRS Journal of Photogrammetry and Remote Sensing*, *175*, 326–339. https://doi.org/10.1016/j.isprsjprs.2021.03.002

Luo, P., Luo, M., Li, F., Qi, X., Huo, A., Wang, Z., He, B., Takara, K., Nover, D., & Wang, Y. (2022). Urban flood numerical simulation: Research, methods and future perspectives. *Environmental Modelling & Software*, *156*, 105478. https://doi.org/10.1016/j.envsoft.2022.105478

Luo, Z., Zhang, Z., Li, W., Chen, Y., Wang, C., Nurunnabi, A. A. M., & Li, J. (2021). Detection of individual trees in uav lidar point clouds using a deep learning framework based on multichannel representation. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–15. https://doi.org/10.1109/tgrs.2021.3130725

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., & Chapman, M. A. (2019). Multi-scale point-wise convolutional neural networks for 3d object segmentation from lidar point clouds in large-scale environments. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 821–836. https://doi.org/10.1109/tits.2019.2961060

Ma, X., Qin, C., You, H., Ran, H., & Fu, Y. (2022). Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint*. https://doi.org/10.48550/arXiv.2202.07123

Madanu, S. C. (2024, November). *A confidence-aware deep learning framework for refining laser-scanned point cloud classification* [Master's thesis]. Delft University of Technology [Available at https://repository.tudelft.nl/record/uuid:e6ab08b9-a4d9-4e9c-86ec-3805631cf998].

Malladi, M. V., Guadagnino, T., Lobefaro, L., Mattamala, M., Griess, H., Schweier, J., Chebrolu, N., Fallon, M., Behley, J., & Stachniss, C. (2024). Tree instance segmentation and traits estimation for forestry environments exploiting lidar data collected by mobile robots. *IEEE International Conference on Robotics and Automation*, 17933–17940. https://doi.org/10.1109/icra57147.2024.10611169

Maltamo, M., Næsset, E., & Vauhkonen, J. (2014). Forestry applications of airborne laser scanning: Concepts and case studies. *Managing Forest Ecosystems*, *27*, 460. https://doi.org/10.1007/978-94-017-8663-8_1

Maturana, D., & Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 922–928. https://doi.org/10.1109/iros.2015.7353481

Mazzini, D., & Schettini, R. (2019). Spatial sampling network for fast scene understanding. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1286–1296. https://doi.org/10.1109/cvprw.2019.00168

Nan, L., & Wonka, P. (2017). Polyfit: Polygonal surface reconstruction from point clouds. *IEEE International Conference on Computer Vision*, 2353–2361. https://doi.org/10.1109/iccv.2017.258

Nazari, M., & Matusiak, B. (2024). Daylighting simulation and visualisation: Navigating challenges in accuracy and validation. *Energy and Buildings*, 114188. https://doi.org/10.1016/j.enbuild.2024.114188

Niemeyer, J., Rottensteiner, F., & Soergel, U. (2013). Classification of urban lidar data using conditional random field and random forests. *Joint Urban Remote Sensing Event*, 139–142. https://doi.org/10.1109/jurse.2013.6550685

Niemeyer, J., Rottensteiner, F., & 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. https://doi.org/10.1016/j.isprsjprs.2013.11.001

Niemeyer, J., Rottensteiner, F., Sörgel, U., & Heipke, C. (2016). Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *41*, 655–662. https://doi.org/10.5194/isprs-archives-xli-b3-655-2016

Ning, X., Ma, Y., Hou, Y., Lv, Z., Jin, H., Wang, Z., & Wang, Y. (2023). Trunk-constrained and tree structure analysis method for individual tree extraction from scanned outdoor scenes. *Remote Sensing*, *15*(6), 1567. https://doi.org/10.3390/rs15061567

Olofsson, K., Holmgren, J., & Olsson, H. (2014). Tree stem and height measurements using terrestrial laser scanning and the ransac algorithm. *Remote Sensing*, *6*(5), 4323–4344. https://doi.org/10.3390/rs6054323

Osco, L. P., De Arruda, M. d. S., Junior, J. M., Da Silva, N. B., Ramos, A. P. M., Moryia, É. A. S., Imai, N. N., Pereira, D. R., Creste, J. E., Matsubara, E. T., et al. (2020). A convolutional neural network approach for counting and geolocating citrus-trees in uav multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *160*, 97–106. https://doi.org/10.1016/j.isprsjprs.2019.12.010

Pađen, I., García-Sánchez, C., & Ledoux, H. (2022). Towards automatic reconstruction of 3d city models tailored for urban flow simulations. *Frontiers in Built Environment*, *8*, 899332. https://doi.org/10.3389/fbuil.2022.899332

Pan, Z., Zhang, N., Gao, W., Liu, S., & Li, G. (2024). Less is more: Label recommendation for weakly supervised point cloud semantic segmentation. *AAAI Conference on Artificial Intelligence*, *38*(5), 4397–4405. https://doi.org/10.1609/aaai.v38i5.28237

Park, Y., & Guldmann, J.-M. (2019). Creating 3d city models with building footprints and lidar point cloud classification: A machine learning approach. *Computers, Environment and Urban Systems*, *75*, 76–89. https://doi.org/10.1016/j.compenvurbsys.2019.01.004

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1912.01703

Peters, R., Dukai, B., Gao, W., & Stoter, J. (2023). 3d bag-geactualiseerd op basis van ahn4. *Geo-Info*, *2023*(2), 30–35. https://research.tudelft.nl/en/publications/3d-bag-geactualiseerd-op-basis-van-ahn4

Philips, D. (2014). Quantifying inflow uncertainties for cfd simulations of the flow in downtown oklahoma city. *Building and Environment*, *78*, 118–129. https://doi.org/10.1016/j.buildenv.2014.04.013

Pu, Y., Xu, D., Wang, H., Li, X., & Xu, X. (2023). A new strategy for individual tree detection and segmentation from leaf-on and leaf-off uav-lidar point clouds based on automatic detection of seed points. *Remote Sensing*, *15*(6), 1619. https://doi.org/10.3390/rs15061619

Puliti, S., Pearse, G., Surovỳ, P., Wallace, L., Hollaus, M., Wielgosz, M., & Astrup, R. (2023). For-instance: A uav laser scanning benchmark dataset for

semantic and instance segmentation of individual trees. *arXiv preprint*. https://doi.org/10.48550/arXiv.2309.01279

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 652–660. https://doi.org/10.1109/cvpr.2017.16

Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., & Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. *IEEE Conference on Computer Vision and Pattern Recognition*, 5648–5656. https://doi.org/10.1109/cvpr.2016.609

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, *30*. https://arxiv.org/abs/1706.02413

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., & Ghanem, B. (2022). Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, *35*, 23192–23204. https://arxiv.org/abs/2206.04670

Ran, H., Liu, J., & Wang, C. (2022). Surface representation for point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18942–18952. https://doi.org/10.1109/cvpr52688.2022.01837

Reche-Martinez, A., Martin, I., & Drettakis, G. (2004). Volumetric reconstruction and interactive rendering of trees from photographs. *ACM SIGGRAPH*, 720–727. https://doi.org/10.1145/1186562.1015785

Redmon, J. (2018). Yolov3: An incremental improvement. *arXiv preprint*. https://doi.org/10.48550/arXiv.1804.02767

Riegler, G., Osman Ulusoy, A., & Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, 3577–3586. https://doi.org/10.1109/cvpr.2017.701

Robert, D., Raguet, H., & Landrieu, L. (2023). Efficient 3d semantic segmentation with superpoint transformer. *IEEE/CVF International Conference on Computer Vision*, 17195–17204. https://doi.org/10.1109/iccv51070.2023.01577

Rosch, E. H. (1973). Natural categories. *Cognitive Psychology*, *4*(3), 328–350. https://doi.org/10.1016/0010-0285(73)90017-0

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., & Breitkopf, U. (2012). The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *1*(1), 293–298. https://doi.org/10.5194/isprsannals-i-3-293-2012

Rottmann, P., Haunert, J.-H., & Dehbi, Y. (2022). Automatic building footprint extraction from 3d laserscans. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *10*, 233–240. https://doi.org/10.5194/isprs-annals-x-4-w2-2022-233-2022

Roynard, X., Deschaud, J.-E., & 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. https://doi.org/10.1177/0278364918767506

Rusu, R. B., Blodow, N., Marton, Z. C., & Beetz, M. (2008). Aligning point cloud views using persistent feature histograms. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3384–3391. https://doi.org/10.1109/iros.2008.4650967

Schonberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. *IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113. https://doi.org/10.1109/cvpr.2016.445

Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., & Leibe, B. (2023). Mask3d: Mask transformer for 3d semantic instance segmentation. *IEEE International Conference on Robotics and Automation*, 8216–8223. https://doi.org/10.1109/icra48891.2023.10160590

Settles, B. (2012). *Active learning*. Springer International Publishing. https://doi.org/10.1007/978-3-031-01560-1

Shan, J., & Toth, C. K. (2018). *Topographic laser ranging and scanning: Principles and processing*. CRC press. https://doi.org/10.1201/9781315154381-1

Sharma, C., & Kaul, M. (2020). Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems*, *33*, 7212–7221. https://arxiv.org/abs/2009.14168

Sharma, M., & Garg, R. D. (2023). Building footprint extraction from aerial photogrammetric point cloud data using its geometric features. *Journal of Building Engineering*, *76*, 107387. https://doi.org/10.1016/j.jobe.2023.107387

Shen, Q. (2024, October). *Plant skeleton extraction and stem-leaf segmentation* [Master's thesis]. Delft University of Technology [Available at https://repository.tudelft.nl/record/uuid:42ae723f-4924-402c-9d8a-992f70f21bf4].

Shi, X., Xu, X., Chen, K., Cai, L., Foo, C. S., & Jia, K. (2021). Label-efficient point cloud semantic segmentation: An active learning approach. *arXiv preprint*. https://doi.org/10.48550/arXiv.2101.06931

Shrestha, D. B., Sharma, R. P., & Bhandari, S. K. (2018). Individual tree aboveground biomass for castanopsis indica in the mid-hills of nepal. *Agroforestry Systems*, *92*, 1611–1623. https://doi.org/10.1007/s10457-017-0109-2

Simonovsky, M., & Komodakis, N. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. *IEEE Conference on Computer Vision and Pattern Recognition*, 3693–3702. https://doi.org/10.1109/cvpr.2017.11

Smith, M. W., & Vericat, D. (2015). From experimental plots to experimental landscapes: Topography, erosion and deposition in sub-humid badlands from structure-from-motion photogrammetry. *Earth Surface Processes and Landforms*, *40*(12), 1656–1671. https://doi.org/10.1002/esp.3747

Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. *ACM SIGGRAPH*, 835–846. https://doi.org/10.1145/1179352.1141964

Stoter, J., Roensdorf, C., Home, R., Capstick, D., Streilein, A., Kellenberger, T., Bayers, E., Kane, P., Dorsch, J., Woźniak, P., et al. (2014). 3d modelling with national coverage: Bridging the gap between research and practice. In *3d*

geoinformation science (pp. 207–225). Springer. https://doi.org/10.1007/978-3-319-12181-9_13

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. *IEEE International Conference on Computer Vision*, 945–953. https://doi.org/10.1109/iccv.2015.114

Su, J., Li, J., Zhang, Y., Xia, C., & Tian, Y. (2019). Selectivity or invariance: Boundary-aware salient object detection. *IEEE/CVF International Conference on Computer Vision*, 3799–3808. https://doi.org/10.1109/iccv.2019.00390

Sun, Y., Li, Z., He, H., Guo, L., Zhang, X., & Xin, Q. (2022). Counting trees in a subtropical mega city using the instance segmentation method. *International Journal of Applied Earth Observation and Geoinformation*, *106*, 102662. https://doi.org/10.1016/j.jag.2021.102662

Sun, Y., Jin, X., Pukkala, T., & Li, F. (2022). Predicting individual tree diameter of larch (larix olgensis) from uav-lidar data using six different algorithms. *Remote Sensing*, *14*(5), 1125. https://doi.org/10.3390/rs14051125

Takikawa, T., Acuna, D., Jampani, V., & Fidler, S. (2019). Gated-scnn: Gated shape cnns for semantic segmentation. *IEEE/CVF International Conference on Computer Vision*, 5229–5238. https://doi.org/10.1109/iccv.2019.00533

Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10781–10790. https://doi.org/10.1109/cvpr42600.2020.01079

Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., & Li, J. (2020). Toronto-3d: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 202–203. https://doi.org/10.1109/cvprw50498.2020.00109

Tang, L., Zhan, Y., Chen, Z., Yu, B., & Tao, D. (2022). Contrastive boundary learning for point cloud segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8489–8499. https://doi.org/10.1109/cvpr52688.2022.00830

Tao, S., Wu, F., Guo, Q., Wang, Y., Li, W., Xue, B., Hu, X., Li, P., Tian, D., Li, C., et al. (2015). Segmenting tree crowns from terrestrial and mobile lidar data by exploring ecological theories. *ISPRS Journal of Photogrammetry and Remote Sensing*, *110*, 66–76. https://doi.org/10.1016/j.isprsjprs.2015.10.007

Tatarchenko, M., Park, J., Koltun, V., & Zhou, Q.-Y. (2018). Tangent convolutions for dense prediction in 3d. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3887–3896. https://doi.org/10.1109/cvpr.2018.00409

Taylor, T. S. (2019). *Introduction to laser science and engineering*. CRC Press. https://doi.org/10.1201/b22159

Thomas, H., Goulette, F., Deschaud, J.-E., Marcotegui, B., & LeGall, Y. (2018). Semantic classification of 3d point clouds with multiscale spherical neighborhoods. *International Conference on 3D Vision*, 390–398. https://doi.org/10.1109/3dv.2018.00052

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). Kpconv: Flexible and deformable convolution for point

clouds. *IEEE/CVF International Conference on Computer Vision*, 6411–6420. https://doi.org/10.1109/iccv.2019.00651

Thomas, H., Tsai, Y.-H. H., Barfoot, T. D., & Zhang, J. (2024). Kpconvx: Modernizing kernel point convolution with kernel attention. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5525–5535. https://doi.org/10.1109/cvpr52733.2024.00528

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 23–30. https://doi.org/10.1109/iros.2017.8202133

Tombari, F., Salti, S., & Di Stefano, L. (2010). Unique signatures of histograms for local surface description. *European Conference on Computer Vision*, 356–369. https://doi.org/10.1007/978-3-642-15558-1_26

Ujang, U., Anton, F., & Rahman, A. A. (2013). Unified data model of urban air pollution dispersion and 3d spatial city models: Groundwork assessment towards sustainable urban development for malaysia. *Journal of Environmental Protection*, *4*(7), 701–712. https://doi.org/10.4236/jep.2013.47081

Varney, N., Asari, V. K., & Graehling, Q. (2020). Dales: A large-scale aerial lidar data set for semantic segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 186–187. https://doi.org/10.1109/cvprw50498.2020.00101

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*. https://arxiv.org/abs/1706.03762

Vu, T., Kim, K., Luu, T. M., Nguyen, T., & Yoo, C. D. (2022). Softgroup for 3d instance segmentation on point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2708–2717. https://doi.org/10.1109/cvpr52688.2022.00273

Wang, D., Liang, X., Mofack, G. I., & Martin-Ducup, O. (2021). Individual tree extraction from terrestrial laser scanning data via graph pathing. *Forest Ecosystems*, *8*, 1–11. https://doi.org/10.1186/s40663-021-00340-w

Wang, J., Chen, X., Cao, L., An, F., Chen, B., Xue, L., & Yun, T. (2019). Individual rubber tree segmentation based on ground-based lidar data and faster r-cnn of deep learning. *Forests*, *10*(9), 793.

Wang, J., Lindenbergh, R., & Menenti, M. (2018). Scalable individual tree delineation in 3d point clouds. *The Photogrammetric Record*, *33*(163), 315–340. https://doi.org/10.1111/phor.12247

Wang, P., Tang, Y., Liao, Z., Yan, Y., Dai, L., Liu, S., & Jiang, T. (2023). Road-side individual tree segmentation from urban mls point clouds using metric learning. *Remote Sensing*, *15*(8), 1992. https://doi.org/10.3390/rs15081992

Wang, P.-S. (2023). Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics*, *42*(4), 1–11. https://doi.org/10.1145/3592131

Wang, P., & Yao, W. (2022). A new weakly supervised approach for als point cloud semantic segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *188*, 237–254. https://doi.org/10.1016/j.isprsjprs.2022.04.016

Wang, S., Suo, S., Ma, W.-C., Pokrovsky, A., & Urtasun, R. (2018). Deep parametric continuous convolutional neural networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2589–2597. https://doi.org/10.1109/cvpr.2018.00274

Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2569–2578. https://doi.org/10.1109/cvpr.2018.00272

Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). Solo: Segmenting objects by locations. *European Conference on Computer Vision*, 649–665. https://doi.org/10.1007/978-3-030-58523-5_38

Wang, X., Liu, S., Shen, X., Shen, C., & Jia, J. (2019). Associatively segmenting instances and semantics in point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4096–4105. https://doi.org/10.1109/cvpr.2019.00422

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, *38*(5), 1–12. https://doi.org/10.1145/3326362

Wei, J., Lin, G., Yap, K.-H., Hung, T.-Y., & Xie, L. (2020). Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4384–4393. https://doi.org/10.1109/cvpr42600.2020.00444

Weinmann, M., Jutzi, B., Hinz, S., & 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. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Weinstein, B. G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., & White, E. P. (2020). Deepforest: A python package for rgb deep learning tree crown delineation. *Methods in Ecology and Evolution*, *11*(12), 1743–1751. https://doi.org/10.1111/2041-210x.13472

Weinstein, B. G., Marconi, S., Bohlman, S., Zare, A., & White, E. (2019). Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. *Remote Sensing*, *11*(11), 1309. https://doi.org/10.3390/rs11111309

West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M., & Iverson, A. E. (2004). Context-driven automated target detection in 3d data. *Automatic Target Recognition XIV*, *5426*, 133–143. https://doi.org/10.1117/12.542536

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300–314. https://doi.org/10.1016/j.geomorph.2012.08.021

Widyaningrum, E., Peters, R. Y., & Lindenbergh, R. C. (2020). Building outline extraction from als point clouds using medial axis transform descriptors. *Pattern Recognition*, *106*, 107447. https://doi.org/10.1016/j.patcog.2020.107447

Wielgosz, M., Puliti, S., Xiang, B., Schindler, K., & Astrup, R. (2024). Segmentanytree: A sensor and platform agnostic deep learning model for tree segmentation

using laser scanning data. *Remote Sensing of Environment*, *313*. https://doi.org/10.1016/j.rse.2024.114367

Wu, C., Bi, X., Pfrommer, J., Cebulla, A., Mangold, S., & Beyerer, J. (2023). Sim2real transfer learning for point cloud segmentation: An industrial application case on autonomous disassembly. *IEEE/CVF Winter Conference on Applications of Computer Vision*, 4531–4540. https://doi.org/10.1109/wacv56688.2023.00451

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., & Zhao, H. (2024). Point transformer v3: Simpler faster stronger. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4840–4851. https://doi.org/10.1109/cvpr52733.2024.00463

Wu, X., Lao, Y., Jiang, L., Liu, X., & Zhao, H. (2022). Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, *35*, 33330–33342. https://arxiv.org/abs/2210.05666

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. *IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920. https://doi.org/10.1109/cvpr.2015.7298801

Wulder, M., Niemann, K. O., & Goodenough, D. G. (2000). Local maximum filtering for the extraction of tree locations and basal area from high spatial resolution imagery. *Remote Sensing of Environment*, *73*(1), 103–114. https://doi.org/10.1016/s0034-4257(00)00101-2

Xi, Z., & Hopkinson, C. (2021). Detecting individual-tree crown regions from terrestrial laser scans with an anchor-free deep learning model. *Canadian Journal of Remote Sensing*, *47*(2), 228–242. https://doi.org/10.1080/07038992.2020.1861541

Xiao, A., Huang, J., Guan, D., Zhan, F., & Lu, S. (2022). Transfer learning from synthetic to real lidar point cloud for semantic segmentation. *AAAI Conference on Artificial Intelligence*, *36*(3), 2795–2803. https://doi.org/10.1609/aaai.v36i3.20183

Xie, S., Gu, J., Guo, D., Qi, C. R., Guibas, L., & Litany, O. (2020). Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *European Conference on Computer Vision*, 574–591. https://doi.org/10.1007/978-3-030-58580-8_34

Xu, D., Zhu, Y., Choy, C. B., & Fei-Fei, L. (2017). Scene graph generation by iterative message passing. *IEEE Conference on Computer Vision and Pattern Recognition*, 5410–5419. https://doi.org/10.1109/cvpr.2017.330

Xu, M., Zhou, Z., Zhang, J., & Qiao, Y. (2021). Investigate indistinguishable points in semantic segmentation of 3d point cloud. *AAAI Conference on Artificial Intelligence*, 3047–3055. https://doi.org/10.1609/aaai.v35i4.16413

Xu, M., Ding, R., Zhao, H., & Qi, X. (2021). Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3173–3182. https://doi.org/10.1109/cvpr46437.2021.00319

Xu, Y., Fan, T., Xu, M., Zeng, L., & Qiao, Y. (2018). Spidercnn: Deep learning on point sets with parameterized convolutional filters. *European Conference on Computer Vision*, 87–102. https://doi.org/10.1007/978-3-030-01237-3_6

Yan, K., Hu, Q., Wang, H., Huang, X., Li, L., & Ji, S. (2021). Continuous mapping convolution for large-scale point clouds semantic segmentation. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5. https://doi.org/10.1109/lgrs.2021.3107006

Yan, X. (2019). Pointnet/pointnet++ pytorch [Accessed: 2022-06-02]. https://github.com/yanx27/Pointnet_Pointnet2_pytorch

Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., & Trigoni, N. (2019). Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1906.01140

Yang, Y.-Q., Guo, Y.-X., Xiong, J.-Y., Liu, Y., Pan, H., Wang, P.-S., Tong, X., & Guo, B. (2023). Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint*. https://doi.org/10.48550/arXiv.2304.06906

Yang, Z., & Wang, L. (2019). Learning relationships for multi-view 3d object recognition. *IEEE/CVF International Conference on Computer Vision*, 7505–7514. https://doi.org/10.1109/iccv.2019.00760

Yang, Z., Ye, Q., Stoter, J., & Nan, L. (2022). Enriching point clouds with implicit representations for 3d classification and segmentation. *Remote Sensing*, *15*(1), 61. https://doi.org/10.3390/rs15010061

Yao, B., Dong, L., Qiu, X., Song, K., Yan, D., & Peng, C. (2024). Uncertainty-guided contrastive learning for weakly supervised point cloud segmentation. *IEEE Transactions on Geoscience and Remote Sensing*. https://doi.org/10.1109/tgrs.2024.3416219

Yazdi, H., Shu, Q., Rötzer, T., Petzold, F., & Ludwig, F. (2024). A multilayered urban tree dataset of point clouds, quantitative structure and graph models. *Scientific Data*, *11*(1), 28. https://doi.org/10.1038/s41597-023-02873-x

Yi, L., Zhao, W., Wang, H., Sung, M., & Guibas, L. J. (2019). Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3947–3956. https://doi.org/10.1109/cvpr.2019.00407

YM., A., C., R., & A., V. (2020). Self-labelling via simultaneous clustering and representation learning. *International Conference on Learning Representations*. https://openreview.net/forum?id=Hyx-jyBFPr

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/mci.2018.2840738

Yu, T., Meng, J., & Yuan, J. (2018). Multi-view harmonized bilinear network for 3d object recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 186–194. https://doi.org/10.1109/cvpr.2018.00027

Yu, Z., Feng, C., Liu, M.-Y., & Ramalingam, S. (2017). Casenet: Deep category-aware semantic edge detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 5964–5973. https://doi.org/10.1109/cvpr.2017.191

Yuan, Y., Xie, J., Chen, X., & Wang, J. (2020). Segfix: Model-agnostic boundary refinement for segmentation. *European Conference on Computer Vision*, 489–506. https://doi.org/10.1007/978-3-030-58610-2_29

Yun, T., Jiang, K., Li, G., Eichhorn, M. P., Fan, J., Liu, F., Chen, B., An, F., & Cao, L. (2021). Individual tree crown segmentation from airborne lidar data using a novel gaussian filter and energy function minimization-based approach. *Remote Sensing of Environment*, *256*, 112307. https://doi.org/10.1016/j.rse.2021.112307

Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, *8*, 58443–58469. https://doi.org/10.1109/access.2020.2983149

Zhang, C., Wan, H., Shen, X., & Wu, Z. (2022). Patchformer: An efficient point transformer with patch attention. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11799–11808. https://doi.org/10.1109/cvpr52688.2022.01150

Zhang, F., Guan, C., Fang, J., Bai, S., Yang, R., Torr, P. H., & Prisacariu, V. (2020). Instance segmentation of lidar point clouds. *IEEE International Conference on Robotics and Automation*, 9448–9455. https://doi.org/10.1109/icra40945.2020.9196622

Zhang, F., Johnson, D. M., Wang, J., & Yu, C. (2016). Cost, energy use and ghg emissions for forest biomass harvesting operations. *Energy*, *114*, 1053–1062. https://doi.org/10.1016/j.energy.2016.07.086

Zhang, J., Lin, X., & Ning, X. (2013). Svm-based classification of segmented airborne lidar point clouds in urban areas. *Remote Sensing*, *5*(8), 3749–3775. https://doi.org/10.3390/rs5083749

Zhang, R., Wang, L., Wang, Y., Gao, P., Li, H., & Shi, J. (2023). Starting from non-parametric networks for 3d point cloud analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5344–5353. https://doi.org/10.1109/cvpr52729.2023.00517

Zhang, Z., Girdhar, R., Joulin, A., & Misra, I. (2021). Self-supervised pretraining of 3d features on any point-cloud. *IEEE/CVF International Conference on Computer Vision*, 10252–10263. https://doi.org/10.1109/iccv48922.2021.01009

Zhao, H., Jiang, L., Fu, C.-W., & Jia, J. (2019). Pointweb: Enhancing local neighborhood features for point cloud processing. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5565–5573. https://doi.org/10.1109/cvpr.2019.00571

Zhao, H., Jiang, L., Jia, J., Torr, P. H., & Koltun, V. (2021). Point transformer. *IEEE/CVF International Conference on Computer Vision*, 16259–16268. https://doi.org/10.1109/iccv48922.2021.01595

Zhao, Y., Wang, J., Li, X., Hu, Y., Zhang, C., Wang, Y., & Chen, S. (2022). Number-adaptive prototype learning for 3d point cloud semantic segmentation. *European Conference on Computer Vision Workshops*, 695–703. https://doi.org/10.1007/978-3-031-25066-8_41

Zhen, M., Wang, J., Zhou, L., Li, S., Shen, T., Shang, J., Fang, T., & Quan, L. (2020). Joint semantic segmentation and boundary detection using iterative pyramid contexts. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13666–13675. https://doi.org/10.1109/cvpr42600.2020.01368

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. *IEEE International Conference on Computer Vision*, 1529–1537. https://doi.org/10.1109/iccv.2015.179

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *IEEE Conference on Computer Vision and Pattern Recognition*, 2921–2929. https://doi.org/10.1109/cvpr.2016.319

Zhou, T., Wang, W., Konukoglu, E., & Van Gool, L. (2022). Rethinking semantic segmentation: A prototype view. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2582–2593. https://doi.org/10.1109/cvpr52688.2022.00261

ACKNOWLEDGEMENTS

I am delighted to have completed my doctoral studies, a milestone that I believe will become a significant stage in my professional career and life. While it is sad to say goodbye to many people I have met in the Netherlands, I wish to express my sincere gratitude to all who have provided warmth, support, and encouragement across the whole journey.

First and foremost, I want to express my heartfelt gratitude and respect to my supervisors, Prof. dr. Jantien Stoter, Dr. Liangliang Nan, and Dr. Julian F.P. Kooij. I am deeply thankful to Jantien for giving me the opportunity to pursue my doctoral studies in the 3D Geoinformation group. Your high-level guidance and rigorous academic support have been invaluable to my research, while your patience, kindness, and genuine care for my personal growth have left a lasting impression on me. I am equally grateful to Liangliang, my daily supervisor. Your dedication, enthusiasm, and commitment to advancing research have been truly inspiring. From you, I have learned fundamental skills in academic thinking, writing, and presentation that will benefit me throughout my future career. My sincere appreciation also goes to Julian for your wisdom, constructive feedback, and kind support. Working and exchanging ideas with you have been enjoyable experiences. Without the guidance, encouragement, and care from all of you, I could not have successfully completed my doctoral studies. Moreover, I would like to express my gratitude to my committee members, Prof. dr. ing. Steffen Nijhuis, Prof. Dr. -Ing. Norbert Haala, Prof. Dr. -Ing. Norbert Pfeifer, Dr. Roderik C. Lindenbergh, and Prof. Dr. -Ing. Uta Pottgiesser, for contributing your valuable time and expertise in evaluating my thesis dissertation.

During the past years in the group of 3D Geoinformation, it has been a pleasure to work and share experiences with my colleagues. I fully enjoyed our coffee breaks, hotpot gatherings, board game nights, casual walks along the riverside, and, in particular, our group trip to Texel Island in autumn 2022. These moments, though small in themselves, have become cherished memories. A sincere thank you goes to Weixiao, Jin, Nail, Zexin, Camilo, Nadine, Zhaiyu, Giorgio, and Xiaoxin for your warm companionship during the pandemic, which was also a personally challenging period for me. I want to extend my thanks to other colleagues in the 3D group, Ivan, Anna, Ravi, Stelios, Hugo, Clara, Ken, Daniele, Lukas, Maarten, Gina, Akshay, Miguel, Siham, Jasper, Amir, and Amy, for contributing to such a collegial and supportive atmosphere. Working and interacting with you has been both enriching and comforting.

Besides, I am grateful to the TU Delft AI Initiative for providing financial support, and to my colleagues at the 3DUU AI Lab, Shiming and Mubariz, for your kindness, wisdom, and all the joyful moments we have spent together. I want to thank Sharath, a master's student with whom I had the pleasure to collaborate. Your talent and dedication to the thesis project have left a strong impression on me. My gratitude also goes to Daan. Your expertise and support were invaluable during the thesis supervision. I will always remember our coffee meetings. Special thanks to the secretaries of Urbanism, Danielle, Martine, Margo, Karin, Romy, and Astrid, whose assistance and kindness have been a consistent support for me.

Throughout this journey, I feel grateful to have met friends who brought me joy and unwavering support. My heartfelt thanks go to Melika, Mey, Teng, Takis, Bill, and Gabo. Our friendship over the past eight years has meant more to me than words can express. I am also thankful to Yuxin, Ziyan, Fenghua, Jing, Qian, Enshan, Yifei, and Shuyu for your kind companionship during my doctoral studies. A very special thank you to two of my dear friends in China, Yu and Shisi, for your constant care and encouragement from so far away. I am sincerely thankful to Yujie, Hongyu, Yuchao, Mengwen, Yini, Yuefei, and Ruby. I deeply cherish all the beautiful moments we shared, and they will remain precious memories to me.

I would like to express my deepest gratitude to my beloved father, Mr. Hongwei Du, and my mother, Mrs. Binian Zhao. Thank you for your unconditional love, endless patience, and constant support throughout my life. Your encouragement and faith in me have always been my greatest source of strength, guiding me through every challenge along the way. I am also grateful to my relatives for standing by me and offering their care and support during difficult times. Your kindness has meant so much to me.

Once again, my heartfelt thanks to everyone who has accompanied me with their help, support, and friendship. The moments we shared have become invaluable treasures in my life, and I will always hold them in fond memory. I wish each of you a fulfilling life, happiness, and continued success in the years to come.

With sincere gratitude,

Shenglan Du

September 2025, Delft

CURRICULUM VITÆ

Shenglan DU

Shenglan was born in Zigui, Hubei province, P.R. China, in 1994. She received her bachelor's degree in Remote Sensing Science and Technology from Wuhan University, Wuhan, China, in 2016. She came to the Netherlands in 2017 and obtained her master's degree in Geomatics at Delft University of Technology, Delft, the Netherlands, in 2019. Besides, she has one year of work experience as a remote sensing specialist at Cobra Groeninzicht, Nijmegen, the Netherlands. From August 2020, she continued her Ph.D. study at the 3D Geoinformation Group, Delft University of Technology, Delft, the Netherlands, under the supervision of Prof. dr. Jantien Stoter, Dr. Liangliang Nan, and Dr. Julian F.P. Kooij. Her research interests include 3D deep learning, point cloud segmentation, and 3D urban scene analysis.

LIST OF PUBLICATIONS

5. **Shenglan Du**, Jantien Stoter, Julian F.P. Kooij, Liangliang Nan. SATree: Structure-aware Tree Instance Segmentation from 3D LiDAR Point Clouds. Urban Forestry & Urban Greening 120 (2026): 129414.

4. Sharath Chandra Madanu[*], **Shenglan Du**[*], Jantien Stoter, Daan van der Heide. RefineNet: A Confidence-aware Deep Online Learning Framework to Refine Real-world Point Cloud Semantic Segmentation. 2026 ISPRS Congress, oral acceptance.

3. Dong Chen, Chenwei Zhu, **Shenglan Du**, Yuliang Wang, Zhen Cao, Mingming Sui, Yiyang Kong, Shengjie Feng, Jiju Peethambaran, Liqiang Zhang. Structure- and Semantics-Aware Mesh Simplification for Generating Lightweight 3D Building Models. Remote Sensing 18.6 (2026): 914.

2. **Shenglan Du**, Nail Ibrahimli, Jantien Stoter, Julian F.P. Kooij, Liangliang Nan. Push-the-Boundary: Boundary-aware Feature Propagation for Semantic Segmentation of 3D Point Clouds. 2022 International Conference on 3D Vision (3DV). IEEE, 2022.

1. **Shenglan Du**, Roderik Lindenbergh, Hugo Ledoux, Jantien Stoter, Liangliang Nan. AdTree: Accurate, Detailed, and Automatic Modelling of Laser-scanned Trees. Remote Sensing 11.18 (2019): 2074.

[*] Equal contribution. Authors are ordered alphabetically.


9 789465 182810