

Mapping forest structure: A 3D deep learning benchmark for ground-based LiDAR

Diego Laino^a, Carlos Cabo^{b,c,*}, Romain Janvier^d, Celestino Ordonez^b, Rodolfo Bolanos^a, Stefan H. Doerr^c, Andrew T. Hudak^e, Cristina Santin^{a,c}

^a Biodiversity Research Institute, CSIC-University of Oviedo-Principality of Asturias 33600 Mieres, Asturias, Spain

^b Department of Mining Exploitation and Prospecting, University of Oviedo, 33600 Mieres, Asturias, Spain

^c Centre for Wildfire Research, Swansea University, Singleton Campus, Swansea SA2 8PP, United Kingdom

^d Independent Consultant, Nancy 54600, France

^e Rocky Mountain Research Station, USDA Forest Service, Moscow, ID 83843, United States

ARTICLE INFO

Keywords:

Forestry
Semantic segmentation
Supervised learning
Remote sensing
Terrestrial Laser Scanning (TLS)
Mobile Laser Scanning (MLS)

ABSTRACT

Detailed characterization of forest structure is critical for applications ranging from production forestry and carbon monitoring to fire risk analysis or habitat and biodiversity assessments. Ground-based LiDAR technologies, such as terrestrial (TLS) and mobile (MLS) laser scanning, enable precise 3D mapping of trees, understorey, and ground, offering rich structural detail. Yet, extracting meaning from these dense, unstructured point clouds remains a persistent challenge that requires reliable and scalable segmentation methods.

Earlier segmentation approaches have struggled to generalize across diverse forest types and acquisition conditions. Recent advances in deep learning (DL), particularly architectures designed for 3D data, offer new opportunities for robust and scalable forest structure modelling using ground-based point clouds. Nevertheless, their implementation in forestry remains limited, partly due to the difficulty of obtaining large, accurately labelled point clouds for training and evaluation.

In this study, we benchmark the performance of six different implementations of four state-of-the-art DL architectures (PointNet, SuperPoint Transformer, Point Transformer V3 and OA-CNN) on the public *Segmented-Forests* dataset, which contains over 850 million labelled points from 14 plots representing a range of coniferous and broadleaf forests scanned with both TLS and MLS. We assess segmentation performance across four ecologically meaningful classes: ground, understorey, stems, and canopy. The best-performing model, Point Transformer V3, achieved an overall mean Intersection over Union (mIoU) of 81.9%, confirming the strong potential of transformer-based architectures for forest point cloud segmentation.

The benchmark analysis considers how segmentation accuracy varies across forest type, sensor, understorey density and forest maturity. We observe clear differences in segmentation difficulty across conditions, with coniferous plots being consistently easier to segment than broadleaf plots, and TLS point clouds yielding higher accuracies than MLS acquisitions. Furthermore, a dense understorey was found to significantly improve class-wise saliency despite increased terrain occlusion. Cross-testing confirmed high model stability when generalized to entirely unseen scenes. Together, these results provide the first comprehensive benchmark of modern DL models for forest segmentation from ground-based LiDAR point clouds and highlight key factors shaping their performance across forest structural diversity.

1. Introduction

The use of ground-based remote sensing point clouds, including those obtained from Terrestrial Laser Scanning (TLS) and Mobile Laser Scanning (MLS), has become increasingly central to the study of forest

structure. These technologies provide high-resolution 3D point clouds that capture intricate structural details often obscured in airborne surveys (Calders et al., 2020) and also overcome limitations of traditional field-based measurements (Disney, 2019). They play a key role in improving biomass estimates (e.g., Kankare et al., 2013; Calderys et al.,

* Corresponding author at: Department of Mining Exploitation and Prospecting, University of Oviedo, Mieres, 33600, Asturias, Spain.

E-mail address: carloscabo@uniovi.es (C. Cabo).

<https://doi.org/10.1016/j.isprsjprs.2026.08.011>

Received 19 December 2025; Received in revised form 23 July 2026; Accepted 15 August 2026

Available online 6 September 2026

0924-2716/© 2026 The Author(s). Published by Elsevier B.V. on behalf of International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

2015; Olschofsky et al., 2016; Gonzalez de Tanago et al., 2018), analysing tree architecture (e.g., García et al., 2015; Li et al., 2017; Liu et al., 2019; Schraik et al., 2021), estimating timber volume and quality (e.g., Kankare et al., 2014; Alvites et al., 2021; Stovall et al., 2023), and characterizing understorey vegetation (e.g., Loudermilk et al., 2009; Olsoy et al., 2014; Alonso-Rego et al., 2020; Tian et al., 2023).

Partitioning the 3D point clouds into informative classes is an important step in utilizing them, as it enables users to focus on the parts that are of interest (Kaijaluoto et al., 2022). Extracting ecological or silvicultural insights from such forest data often requires the classification of individual points into meaningful categories, such as stems, branches, understorey vegetation, or ground, a task commonly referred to as semantic segmentation (Kulicki et al., 2024). However, the segmentation of 3D point clouds poses unique challenges when compared to 1D or 2D data, primarily due to their unstructured and unordered nature, irregular sampling density, and lack of a grid topology (Hackel et al., 2016; Bello et al., 2020). Moreover, the increasing resolution (up to millimetre level) and size of ground-based point clouds further expand their range of potential applications but also pose major challenges for efficient segmentation (Lines et al., 2022).

In addition to the challenges that are inherent to 3D data, the semantic segmentation of forest point clouds comes with additional difficulties particular to the nature of forests (Cherlet et al., 2026). In this regard, forest point clouds are characterized by a lack of planar or regular primitives and great geometric complexity (Xiang et al., 2025). For instance, a stem can be a straight cylinder but can also bifurcate and sprawl, and canopies of different tree species vary vastly in shape. Additionally, the boundary between classes is often physically continuous or geometrically identical (e.g., a small twig on a tree can look exactly like a small twig in a shrub). This makes much harder for a model to learn a “global shape” for a class. This difficulty is compounded by extreme intra-class variance, where no two biological structures are identical, and by significant density gradients caused by severe self-occlusion (e.g., within-canopy occlusion) (Kaijaluoto et al., 2022; Lines et al., 2022). All of these factors make semantic segmentation of forest scenes an extremely challenging task.

Earlier approaches to segmenting forest point clouds can be broadly divided into two categories: those that do not involve learning, and those that rely on learning. Non-learning methods include rule-based approaches that assign labels based on explicit geometric conditions (e.g., height thresholds, surface normals) (Ma et al., 2016), and connectivity-based methods such as region growing or graph-based clustering (Tao et al., 2015; Wang et al., 2018; Vicari et al., 2019; Hui et al., 2021). These strategies range from simple to highly elaborate, and can be used in combination to improve performance (Wang, 2020). While they can work well in specific settings, they typically require extensive manual tuning and struggle to generalize across different forest types or acquisition conditions (Kaijaluoto et al., 2022; Kulicki et al., 2024).

In contrast, 3D learning-based methods aim to automate semantic segmentation by deriving discriminative structural patterns from the point cloud geometry, though they differ in how these patterns are identified and the degree of supervision required (LeCun et al., 2015). Supervised learning utilizes labelled datasets to explicitly establish the relationship between geometry and semantic classes. Supervised approaches (like random forests or support vector machines) typically perform segmentation by processing handcrafted features, that is, numerical descriptors explicitly designed by the user and representing specific characteristics of each point and its local neighbourhood (Cabo et al., 2019). While effective, the performance of these models is heavily dependent on the quality of feature engineering and the availability of large amounts of labelled data, which are logistically difficult to produce in forest environments. Alternatively, unsupervised learning approaches (Liu et al., 2023; Rozenberszki et al., 2024; Zhang et al., 2023; Chen et al., 2024; Zhang et al., 2024) rely on data parametrization rather than explicit annotations to identify geometric primitives. Nevertheless, they

still require labelled data to be evaluated and, while they have shown significant promise in specific forestry-related tasks (e.g., leaf-vs-wood segmentation; Wang et al., 2019; Burmeister et al., 2025), their ability to generalize across full-plot, multi-class segmentation in diverse forest types remains an emerging area of research.

More recently, deep learning (DL) methods have emerged as the most powerful and flexible solution for 3D point cloud segmentation (LeCun et al., 2015). Their advances come with trade-offs, though. DL architectures are typically computationally intensive, demanding high-end CPU/GPU resources, and they require vast quantities of accurately labelled training data to reach peak performance (LeCun et al., 2015; Bello et al., 2020; Camuffo et al., 2022). Furthermore, their ‘black-box’ nature offers limited user control over feature estimation compared to classical methods, which can hinder the interpretability of results (Şahin et al., 2025). However, unlike classical techniques, deep networks can learn feature representations directly from raw 3D coordinates and their spatial context (Vaswani et al., 2017), while also allowing the integration of handcrafted features that may provide complementary information. This dual capacity enables DL methods to combine data-driven feature extraction with domain-specific knowledge, often resulting in improved segmentation performance (Kulicki et al., 2024). In this context, a substantial body of work has been dedicated to “wood-leaf” discrimination (e.g., Wang et al., 2019; Wu et al., 2020; Han & Sánchez-Azofeifa, 2022; Alvites et al., 2023; Dai et al., 2023; Jiang et al., 2023; Kim et al., 2023; Bryson et al., 2024; Lu et al., 2025) and tree instance segmentation (Straker et al., 2023; Henrich et al., 2024; Wielgosz et al., 2024; Xiang et al., 2024; Burmeister et al., 2025; Xi and Degenhardt, 2025; Cherlet et al., 2026), a task closely related to, and sometimes overlapping with semantic segmentation, since the fundamental goal is still to decompose the forest scene. Some studies have also put the focus on specific tasks as coarse woody debris detection (Xi et al., 2023).

Overall, however, few studies have explored multi-class forest segmentation schemes at the plot level (Krisanski et al., 2021; Kaijaluoto et al., 2022; Shen et al., 2022; Ma et al., 2023; Oviedo de la Fuente et al., 2024). Most of these approaches incorporate handcrafted features, and they have shown that these descriptors improve segmentation performance in complex forest structures. However, results vary substantially across datasets, segmentation schemes and acquisition settings, and no single procedure has emerged as universally superior (Kulicki et al., 2024). Thus, despite the rapid advancement of 3D DL, its application in plot-level, multi-class segmentation of forest scenes remains relatively limited (Kulicki et al., 2024). Furthermore, existing studies utilize seminal but dated 3D DL architectures as PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b) and DGCNN (Wang et al., 2018), leaving modern Transformer-based (Vaswani et al., 2017) and sparse convolutional networks (Liu et al., 2015) untested in forest environments. And, although DL is becoming the dominant paradigm for point cloud segmentation in forestry, the field still lacks systematic benchmarks and robust comparative evaluations across architectures. This is, in part, due to the logistical and physical constraints inherent in forest data acquisition, which result in a scarcity of high-quality training data and studies using private datasets. This fragmentation makes it nearly impossible to compare results across publications and underscores the need for a standardized evaluation protocol using open-source data (Lines et al., 2022; Kulicki et al., 2024). Recent benchmark initiatives in tree instance segmentation (Puliti et al., 2023; Burmeister et al., 2025; Cherlet et al., 2026) have begun to address this by providing comparative evaluations across varied models and datasets. However, we are only beginning to identify the key architectural limitations and ‘failure modes’ against forest heterogeneity, which limits both our understanding of the real-world applicability and the development of practical, generalizable solutions.

The release of *SegmentedForests* (Laino et al., 2025), a publicly available benchmark dataset that comprises 14 forest plots spanning a wide range of forest types and acquisition conditions (TLS and MLS), offers an opportunity to address this gap. Researchers now have access

to a large, diverse and open dataset specifically designed for semantic segmentation of forest plots in ground-based point clouds. Yet, no study to date has provided a systematic, head-to-head comparison of modern deep learning models. Here, we present the first benchmark of this kind, combining recent architectures with open data to establish a new reference point for forest point cloud segmentation.

In this article, we evaluate the performance of six implementation of four state-of-the-art DL architectures for the semantic segmentation of ground-based forest point clouds: two implementations of PointNeXt, two implementations of SuperPoint Transformer (SPT), one implementation of Point Transformer V3 (PTV3), and one implementation of Omni-Adaptive Convolutional Neural Networks (OA-CNN). We aim to quantify their ability to classify points into four ecologically and functionally meaningful structural categories in the context of forestry: ground, understorey, stem, and canopy. This class structure reflects the main vertical and functional strata of forest ecosystems, each of which plays a distinct role in ecological processes and forest management. To ensure a fair and reproducible comparison, we implement a standardized benchmark where all architectures are trained and tested on a level playing field.

In addition, we investigate whether model performance differs across a spectrum of forest heterogeneity using a suite of plot-level descriptors. We disaggregate our results in forest type (Coniferous vs Broadleaf), acquisition technology (MLS vs TLS), understorey density (Low vs High) and forest maturity (Young/Mid vs Mature). Finally, we assess the generalizability of the results to unseen forests via a leave-one-out cross-testing protocol. With these analyses we aim to move beyond accuracy metrics and provide insights into how specific forest configurations affect the spatial consistency and generalizability of modern DL architectures.

2. Materials and methods

2.1. Architectures

The benchmarked DL architectures include PointNeXt (Qian et al., 2022), SPT (Robert et al., 2023), PTV3 (Wu et al., 2024), and OA-CNN

(Peng et al., 2024) (Fig. 1). Each model was selected based on its relevance and performance in the field of point cloud processing and semantic segmentation. All four models are built around the same general architecture: a U-Net-like encoder-decoder structure, commonly used in segmentation tasks. The encoder (the contracting path) gradually trades spatial resolution for richness in the feature/latent space, and the decoder (the expanding path) remaps these features back to higher resolution through a mix of upsampling techniques and information from the corresponding encoder level (via skip connections). However, they differ in three important aspects that affect how they process spatial information (Fig. 1). First, they reorganize the point cloud into small, meaningful groups (or “pre-structure” it) in different ways before learning from it: PointNeXt uses point subsampling, SPT defines a set of Superpoints, PTV3 serialized patches of voxels with space-filling curves and OA-CNN uses a voxel grid. Second, they vary in how they learn information about each point’s position in space (e.g., if they incorporate positional encoding and in which stage it is done). And third, they differ in how they capture the surrounding context of each point: while PointNeXt use a multiscale PointNet architecture, SPT and PTV3 use attention (transformer) layers (Vaswani et al., 2017) to adaptively learn relationships and OA-CNN uses a lightweight alternative to attention layers called Adaptive Relation Convolution (ARConv) layers.

PointNeXt

PointNeXt is a high-performance baseline for 3D point cloud processing, extending the concept of convolutional neural networks (CNNs) to irregular and unordered point sets (Qian et al., 2022). Introduced in 2022, it builds upon the principles of the popular PointNet++ (Qi et al., 2017a). PointNet++ extends PointNet (Qi et al., 2017b) by introducing a hierarchical grouping of points (Set Abstraction Layers, Fig. 1), which progressively abstracts increasingly larger local regions along the hierarchy, balancing local detail with global context. PointNet++ applies the PointNet architecture recursively on a nested partitioning of the input point set, which is achieved through a combination of farthest point sampling (FPS), neighbourhood grouping (via either k-Nearest-Neighbours—k-NN—, or search radius), and local feature aggregation.

PointNeXt itself expands upon the PointNet architecture by

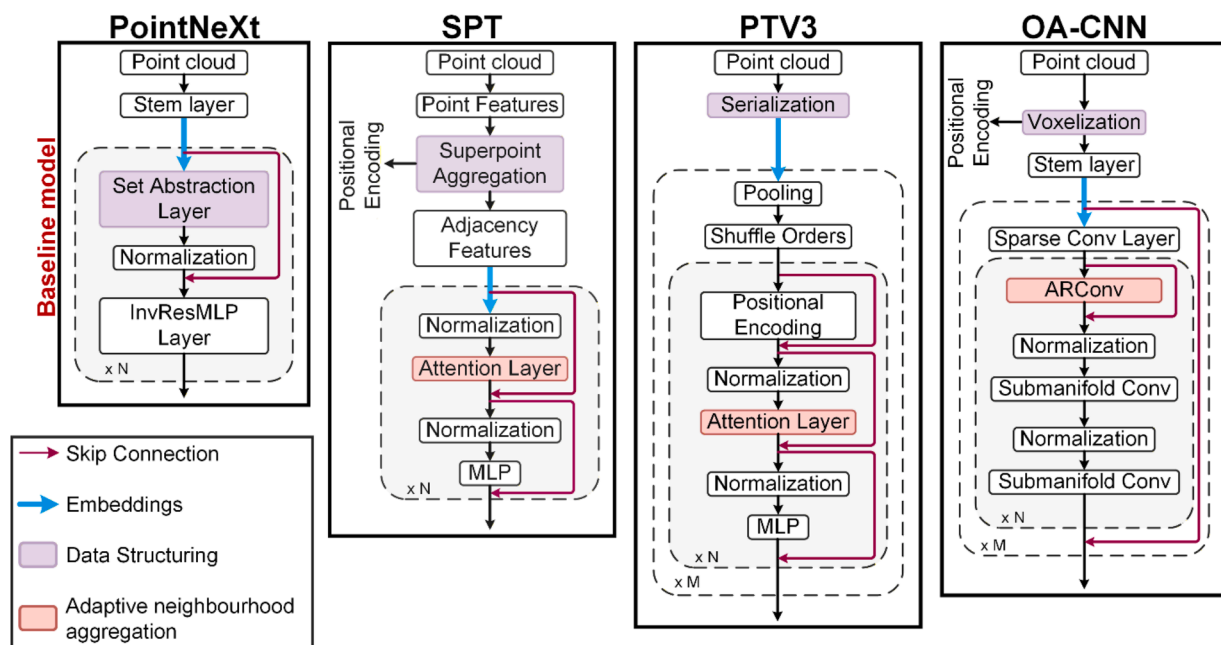


Fig. 1. Summary of the four point-based deep learning model architectures evaluated in this study for structure-aware forest segmentation from ground-based LiDAR point clouds. All follow a U-Net-like encoder-decoder structure.

incorporating recent advances introduced by modern CNN architectures. Namely, an Inverted Bottleneck design with Residual Connections (InvResMLP; He et al., 2016) (Fig. 1) to improve gradient flow and enable deeper networks (deeper models are expected to perform better; Bello et al., 2021; Li et al., 2021; Zhai et al., 2022) and Separable Multi-Layer Perceptrons (MLPs) (Qian et al., 2021) to reduce computation and improve pointwise feature extraction.

PointNeXt has demonstrated strong performance in segmentation tasks on open computer vision datasets (Qian et al., 2022) and incorporates modern architectural solutions to the two main challenges associated with processing large-scale 3D point clouds: the ability to capture both local and global information to decide what semantic label should be assigned to a point, and scalability. Additionally, its modular design also makes it adaptable for incorporating handcrafted features. However, its architecture does not incorporate mechanisms to adaptively learn spatial relationships as attention layers or ARConv layers (Fig. 1). For these reasons, PointNeXt is used in this study as a baseline for benchmarking newer architectures.

SuperPoint Transformer

SPT is a state-of-the-art deep learning model introduced in 2023 (Robert et al., 2023) and designed for efficient and scalable processing of large-scale 3D point clouds. SPT builds upon the seminal SuperPoint Graph method (Landrieu and Simonovsky, 2018) but adds transformer-based self-attention mechanisms (Vaswani et al., 2017) into the architecture. SPT incorporates a pre-processing step to over-segment the input point cloud into Superpoints (Landrieu and Simonovsky, 2018) (Fig. 1), which are aggregations of spatially and semantically consistent points. This way, SPT reduces computational complexity by allowing the network to operate on a smaller set of representative, hierarchical regions instead of the whole set of points.

To partition the point cloud into Superpoints, an initial k-NN graph and a set of handcrafted features are computed. Then, this graph is split into a hierarchical partition using an unsupervised geometric partition technique (Landrieu and Simonovsky, 2018). The resulting components are dubbed Superpoints. A key property of the Superpoints is that they share adjacency relationships with both Superpoints across and within different partition levels. Then, for each Superpoint, geometric features are computed (Fig. 1). To do so, for coarser Superpoints with no parent, features are learned from the positions and the handcrafted features of the points it contains using a MLP. Feature information from Superpoints is propagated within hierarchical levels. To propagate information within a level, the features learned from the set of Superpoints within that level are aggregated via max pooling and encoded using attention layers (Fig. 1). Finally, information within sets of neighbouring Superpoints is propagated across hierarchical levels in a similar fashion.

The simplification of the point cloud to this set of Superpoints and the inclusion of self-attention through transformer modules provides the ability to model long-range relationships and interactions between points while reducing computational overhead. This allows SPT architecture to achieve high accuracy while being suitable to process large-scale point clouds (Robert et al., 2023). These properties make SPT potentially well suited for plot-level forest segmentation, where ecologically meaningful structures often span broad spatial extents and require the modelling of long-range spatial relationships.

PTV3

Similarly to PointNeXt, PTV3 expands predecessor architectures of deep learning models to address the challenges of capturing both local geometric details and global contextual relationships while trying to remain scalable (Wu et al., 2024). The original Point Transformer (Zhao et al., 2021) architecture pioneered the use of self-attention mechanisms

through Transformer layers in 3D point cloud processing. Point Transformer V2 (Wu et al., 2022) refined the implementation, improving computational efficiency and accuracy. PTV3 further advances this lineage thanks to optimizations in attention mechanisms and neighbourhood aggregation (Wu et al., 2024). This is especially relevant for forest plot data, where structural complexity and spatial extent require both scalability and contextual sensitivity.

The original Point Transformer introduces local self-attention mechanisms to model the relationships between points in a cloud. This mechanism allows the model to dynamically learn contextual interactions within neighbourhoods, similar to the approach seen in SPT but applied directly to local k-NN neighbourhoods rather than Superpoints. What sets Point Transformer apart from previous transformer-based architectures is that it includes a Vector Attention (VA) layer instead of a Scalar Attention layer (Vaswani et al., 2017). This VA embeds positional encoding about the spatial difference between neighbouring points through a MLP and a measure of relationship between points that is computed via a relation function. The VA layers in Point Transformer, however, suffer from scalability problems (Wu et al., 2022). As the network goes deeper, the number of its VA weights increases drastically. In order to overcome these limitations, Point Transformer V2 introduces Grouped Vector Attention (GVA). In GVA, VA weights are shared within groups of points, which greatly reduces the number of parameters of the model, improving its overall efficiency.

Finally, to further leverage the power of raw scale, PTV3 introduces a series of performance improvements to PTV2. These mainly consist of transforming the point cloud from an unstructured format to a serialized structure using a space-filling curve (Hilbert, 1935; Morton, 1966) (Fig. 1), and modifications to neighbourhood, attention and positional encoding computations to fit this new paradigm. Instead of grouping points using k-NN neighbourhoods, the voxelized point cloud is segmented regularly across the space-filling curves in patches. The attention mechanism is then applied to those patches (Fig. 1), and the positional encoding from PTV2 is substituted by a modern, more efficient conditional positional encoding (CPE) (Chu et al., 2021). With these improvements PTV3 significantly increases computational speed ($3 \times$ faster) and reduces memory demands ($10 \times$ less) compared to PTV2 without compromising accuracy, making it particularly well-suited for large-scale semantic segmentation tasks.

OA-CNN

OA-CNN (Peng et al., 2024) is a framework for 3D point cloud segmentation that adopts a different approach and bases its architecture on sparse convolution modules (Fig. 1). These are easy to apply, lightweight and leverage the benefits of (locally) structured data (Peng et al., 2024). However, they lack the adaptivity of the self-attention mechanism from point cloud transformers, which limits their accuracy when compared to the latter. To compensate for this, OA-CNN incorporates an ARConv module (Fig. 1), which is lighter than Transformer modules, and that also serves the purpose of being able to adapt the receptive field size based on local context. This allows OA-CNN to extract spatial information from groups of points very efficiently, achieving up to $5 \times$ greater speed than transformer-based networks in open 3D datasets (Peng et al., 2024).

In sparse CNNs, point clouds are voxelized (Fig. 1) and then 3D convolutions kernels are operated onto the voxels. Submanifold 3D Sparse Convolutions are then applied to the voxels in sparse CNNs (Fig. 1). To achieve this, a voxel grid size and a convolution kernel are defined. Either fine grid sizes or large kernels produce an intolerably heavy increase in the number of parameters of standard sparse CNNs, which is one of their main limitations (Chen et al., 2023). OA-CNNs addresses this introducing a novel 3D sparse convolution-based module, the ARConv. This module operates in multi-scale pyramid grids with

K grid sizes, weighting and fusing convolution outputs across scales. The simplicity and speed of OA-CNN makes it especially attractive in scenarios where processing constraints are critical, such as large forest inventories or repeated scanning campaigns.

2.2. Dataset

All the experiments described in this article were conducted using *SegmentedForests* (Laino et al., 2025), a public dataset of manually labelled data that has been designed for benchmarking semantic segmentation of ground-based point clouds in forest environments. The dataset covers forest plots with diverse structure, composition, and acquisition conditions (Table 1). Specifically, the point clouds were acquired using both TLS and MLS technologies in a range of temperate forest ecosystems, capturing broad variation in species, tree density, canopy structure, understorey coverage and scanning geometry. Overall, the dataset comprises 14 forest plots that encompass more than 1600 trees across 16 different species and that cover 2.3 ha across four countries (Austria, Spain, United Kingdom, and United States). More detailed information regarding the geographic location, data acquisition protocols, and point cloud pre-processing steps is provided in the original dataset publication (Laino et al., 2025).

The original data has 16 manually labelled semantic classes but for the experiments conducted here these were aggregated into four broader categories (Fig. 2): 0 – “Ground” is unchanged from the original dataset. 1 – “Understorey”, includes “shrub” and the four “understorey tree” classes present in the original dataset. 2 – “Stem”, includes both “stem” and “standing dead stem”. 3 – “Canopy” is unchanged from “branches + leaves”. These aggregate classes align with the structural components of interest presented in the Introduction.

The remaining seven *SegmentedForests* classes were aggregated into an additional class 4: “Other” that aggregates low-frequency or structurally ambiguous elements, including “dead down wood”, “vine”, “people”, “rocks”, “stakes”, “stump”, and “other”. The aggregate “Other” class was excluded from the experiments since it represents only a small fraction of the point cloud (~3.5% of all points) and individual sub-classes within it appear too sparsely to allow for the learning of representative geometric features. Additionally, when aggregated, the class is geometrically incoherent; it lacks a consistent structural signature, which would introduce significant noise into the feature-learning process.

2.2.1. Data partitions

To evaluate the robustness of the deep learning models across varying forest heterogeneity, the dataset was partitioned into categorical groups based on plot descriptors presented in Table 1 and into structural archetypes derived from a Principal Component Analysis (PCA) of the plot structural descriptors (Fig. 3). The PCA identified two primary axes of variability that define “structural archetypes”. The first one (PC1, 27.6% explained variance) is determined by Understorey cover (the proportion of the plot area covered by shrubs and understorey trees), Ground (%) and Understorey (%) (the proportion of points with “Ground” and “Understorey” labels, respectively). It therefore is an indicator of the variability in understorey density. In this regard, plots with higher understorey density feature higher understorey cover and a higher proportion of points labelled as “understorey”. In turn, the understorey occludes the ground, which is denoted by a decreasing amount of “ground” points. The second axis (PC2, 24.8% explained variance) is determined by Plot Density (trees / ha), dominant DBH, Canopy cover (the proportion of the plot area covered by the canopy) and Stem (%) (the proportion of points with “Stem” label). In this axis, dominant tree diameter at breast height (DBH) is inversely proportional to tree density, canopy cover and “stem” points. Thus, it can be understood as a gradient in “Forest Maturity”, where more mature plots have thicker trees with larger canopies and lower tree densities (McGee et al., 1999).

Table 1
Summary of plot characteristics of the SegmentedForests dataset, for more details see Laino et al. (2025). The last four columns (“Ground (%)”, “Understorey (%)”, “Stem (%)” and “Canopy (%)”) are the percentage of points that belong to each of the aggregated classes. BL: Broadleaf. CO: Coniferous. Q. pyr: *Quercus pyrenaica*. Q. syl: *Quercus sylvatica*. P. pin: *Pinus pinaster*. E. glob: *Eucalyptus globulus*. P. abi: *Picea abies*. F. exc: *Fraxinus excelsior*. A. alb: *Abies alba*. Q. rob: *Quercus robur*. P. pon: *Pinus ponderosa*. P. pal: *Pinus palustris*.

Plot Forest Type	Main species	Sensor	Area (m ²)	Point Density (1000 pts / m ²)	Density (trees / ha)	Dominant TH (m)	Dominant DBH (cm)	Structural Descriptors							
								Canopy Cover (%)	Understorey Cover (%)	Slope (%)	Ground points (%)	Understorey points (%)	Stem points (%)	Canopy points (%)	
1	BL	Q. pyr.	MLS	650	32	25.5	26	88	38	17	23	8	29	40	
2	BL	F. syl.	MLS	2100	11	27	88	95	1	14	28	0	8	64	
3	CO	P. rad.	MLS	1560	27	17	43	86	40	30	10	16	22	50	
4	BL	E. glo.	MLS	930	35	22	28	69	81	26	5	38	20	33	
5	CO	P. Pin.	MLS	680	46	9	23	82	32	1	18	11	24	47	
6	BL	F. syl.	TLS	1300	59	27.5	40	90	48	14	24	7	19	43	
7	CO	P. abi.	TLS	1410	42	22.5	38	85	18	10	31	5	28	33	
8	BL	F. syl. & F. exc.	TLS	1390	202	26	33	93	2	11	51	1	12	27	
9	CO	P. abi. & F. syl.	TLS	1460	50	38	73	92	4	7	37	0	27	34	
10	CO	A. alb. & P. abi.	TLS	1430	30	35	70	75	55	7	12	16	28	40	
11	BL	Q. rob.	MLS	900	21	19	62	96	29	10	16	7	10	63	
12	CO	P. pon.	TLS	3590	18	20	55	18	51	1	36	19	2	37	
13	CO	P. pon.	TLS	3230	19	20	50	24	32	1	30	21	3	43	
14	CO	P. pal.	TLS	2490	8	24.5	50	45	65	3	4	16	7	69	

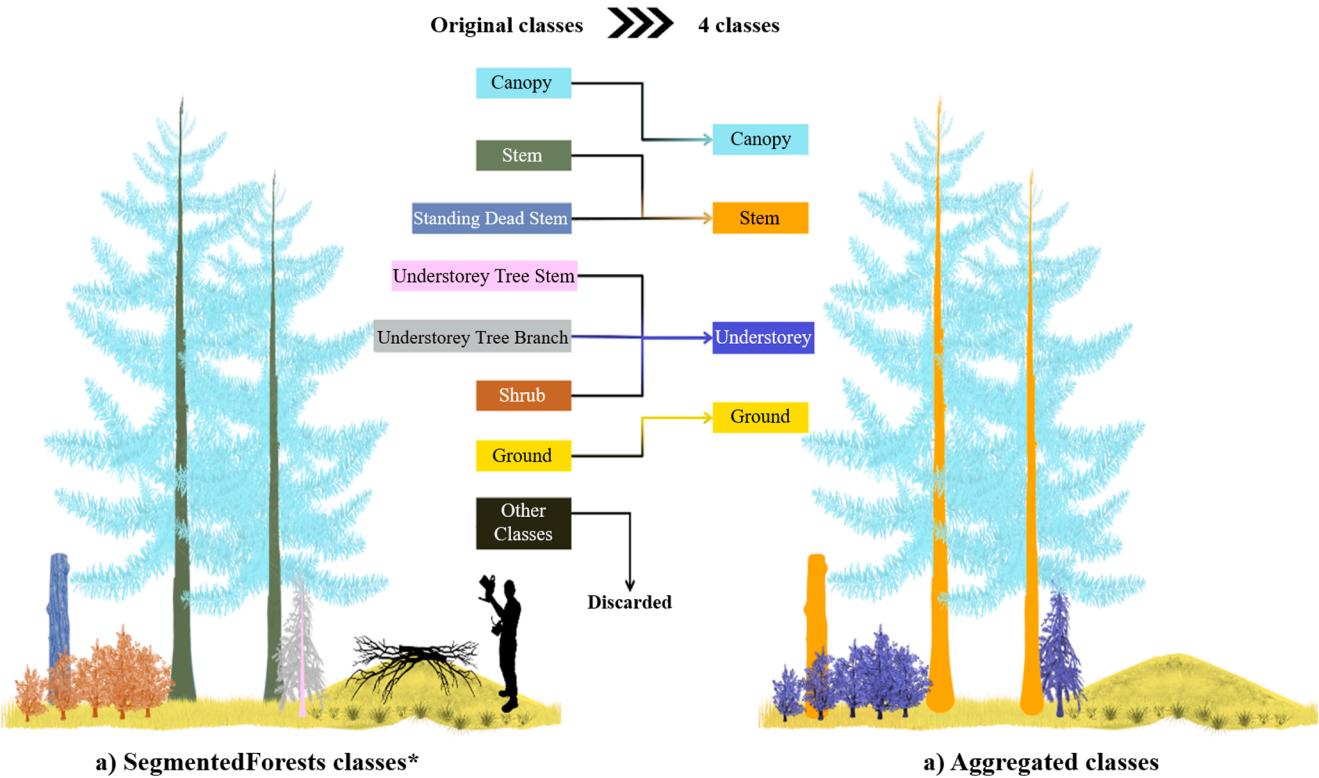


Fig. 2. Original classes in the SegmentedForests dataset* (a) and class aggregation into final segmentation scheme used in this study (b). *Note that not all the 16 original classes are illustrated here, for full details see Laino et al. (2025).

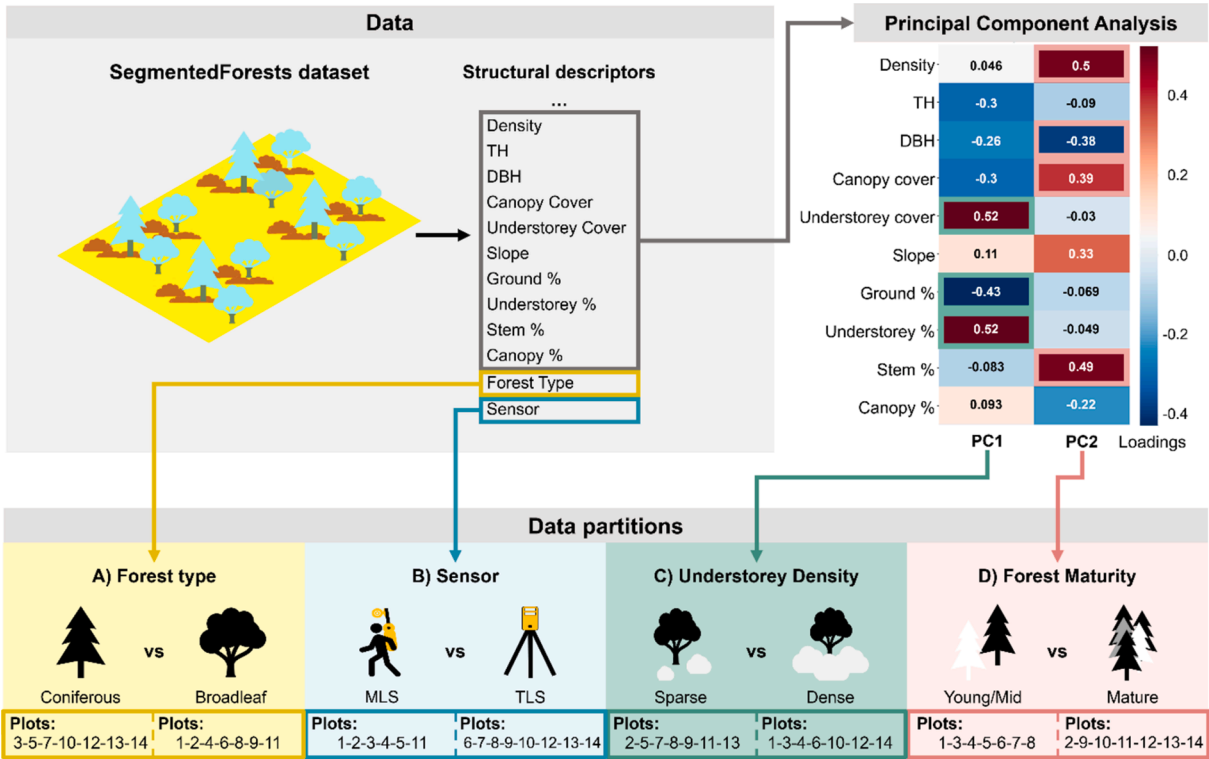


Fig. 3. Data partitions, showing what plots are included in each group. Partitions (A) “Forest Type” and (B) “Sensor” are directly based on categorical descriptors shown in Table 1. Partitions (C) “Understorey Density” and (D) “Forest Maturity” are based on the PCA of numerical plot structural descriptors. These descriptors are also presented in Table 1.

Based on the categorical variables directly available in the dataset description, the plots were categorized by A) Forest Type and B) Sensor, and based on the PCA the plots were partitioned by C) Understorey Density and D) Forest Maturity (Fig. 3):

- A) Forest type: the plots were categorized by forest type into “coniferous” or “broadleaf” (depending on the dominant tree species; Table 1). In the original *SegmentedForests* dataset, plots are labelled as either “coniferous,” “broadleaf,” or “mixed.” For this experiment, plots originally labelled as “mixed” were reassigned to the group corresponding to their dominant tree type. As a result, plots 7, 9, and 10 were treated as “coniferous” based on the higher relative proportion of conifers.
- B) Sensor: to evaluate the impact of the acquisition mode, plots were divided into Mobile Laser Scanning (MLS) and Terrestrial Laser Scanning (TLS) groups (Table 1). This distinction allows for the assessment of model robustness against varying point densities and sensor-specific occlusion patterns.
- C) Understorey Density: the plots were partitioned by using a threshold of 37.5% understorey cover, representing the midpoint of the middle cover-abundance classes of the Braun-Blanquet scale (Wikum & Shanholtzer, 1978). This threshold is the boundary between the three lower and three upper cover classes defined in this scale and separates plots where the understorey is a subordinate structural element (“Low Density”) from those where it becomes a dominant component (“High Density”) of the forest profile.
- D) Forest Maturity: Stands were partitioned based on a dominant DBH threshold of 50 cm. In temperate forests, 50 cm / > 20 in. is a common criterion for defining “mature” or “old-growth” structural stages (Buongiorno et al., 1994; McGee et al., 1999). Thus, the plots were divided in “Young/Mid” vs “Mature”. This maturity threshold also effectively captures the stand's self-

thinning dynamics, as mature plots exhibit lower stem densities (thus larger, more distinct geometric primitives for stems) compared to younger, more crowded stands.

2.3. Processing and implementation

The following subsections describe the workflow applied to process the *SegmentedForests* dataset and to implement the DL models. Data preparation involved aggregating the original *SegmentedForests* classes into the final segmentation scheme, voxelizing the point clouds and computing the set of features. For model implementation, the experimental protocol was kept fixed across models in terms of data splits, input features, computational constraints, and evaluation metrics of model performance. This overall workflow is summarized in Fig. 4.

2.3.1. Voxelization

Feeding the point clouds directly to the models exceeded the memory capacity and computational efficiency limits of current state-of-the-art architectures, necessitating a downsampling approach. To make the process computationally efficient, all point clouds were first voxelized (Fig. 4) to reduce their spatial resolution while preserving essential structural information. Voxelizing the point clouds also has the added benefit of reducing sensitivity to noise points caused by scanner errors and registration errors (Putman et al, 2018). Voxelization was done using the in-house Python package *dendroptimized* (Janvier and Cabo, 2024). Semantic labels were assigned to each voxel using a majority voting scheme, where a voxel is assigned the label of the most frequent class among the points falling within its boundaries. This majority voting approach is necessary for computational tractability but introduces a degree of discretization error, which may slightly influence the final point-level accuracy metrics. Nevertheless, performing inference in voxelized point clouds means predictions are made at the voxel level, rather than the point level. In order to predict point labels, voxel-

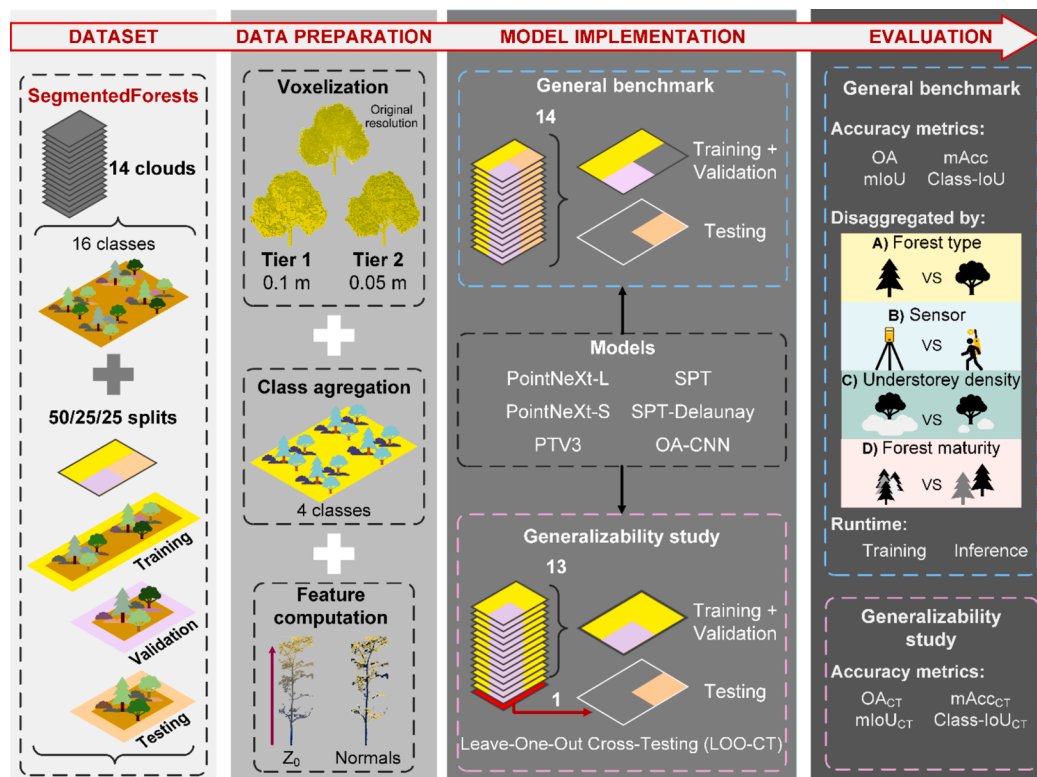


Fig. 4. Schematic overview of the data processing and model implementation workflow.

level predictions needed to be remapped back to the original points. To enable this, each point in the original cloud was assigned a voxel ID during pre-processing, corresponding to the voxel it was aggregated into Cabo et al. (2014). After inference, the predicted label of each voxel was propagated to all original points sharing its ID. This ensured that accuracy metrics were computed consistently at point-level resolution, even though the models operated on voxelized inputs.

2.3.2. Feature computation

All experiments except SPT-based ones were conducted using the following features: surface normal vectors (normals) at each point and the height above ground (Z_0) (Fig. 4). These normals were computed in Python at 0.5 m search radius. Z_0 was calculated using the height normalization module based on Cloth Simulation Filter (CSF; Zhang et al., 2016) that is implemented in 3DFin v0.5.0 (Laino et al., 2024), a widely used virtual forest inventory tool developed by the authors of this study. For SPT-based models normals were dropped in favour of their built-in set of k-NN-based handcrafted features.

2.3.3. Benchmarking setup

The experimental framework was structured into a two-tier general benchmark (Fig. 4) to assess both architectural fairness and geometric scalability:

Tier 1: All architectures were first compared using a common 0.1 m voxel resolution. This represents a compromise between geometric fidelity and computational feasibility, preserving primary forest structures while remaining tractable.

Tier 2: The benchmark was repeated at a 0.05 m resolution to evaluate the impact of increased geometric detail. This voxel size resolves smaller patches of understorey and thinner trees; however, is computationally much more expensive.

Although voxel size was standardized at the input level, we acknowledge that internal architectural mechanisms (e.g., superpoint oversegmentation in SPT or FPS sampling in PointNeXt) result in varying levels of internal data quantization; these implications are discussed in Section 4.3 Runtime Trade-offs. Model performance was further disaggregated at both voxel sizes by the data partitions defined in Section 2.2.1 Data Partitions to explore the effect of forest structure in segmentation quality. To standardize the results, we adhered to the data splits that are available in the *SegmentedForests* dataset for model benchmarking (Laino et al., 2025). The dataset includes a predefined 50/25/25 train/validation/test split (Fig. 4) which contain, respectively, ~398, ~245 and ~ 205 million points. This split was used consistently across all experiments done in this study to facilitate comparability of results. This ensures that performance variations are attributable to architectural differences rather than differences in data sampling.

2.3.4. Generalizability study

In addition to the general benchmark, we implemented a Leave-One-Plot-Out Cross-Testing (LOO-CT) protocol of 14 iterations (one per each plot) to assess the spatial generalizability of the results (Fig. 4). For each iteration, one forest plot was completely withheld from training. For the remaining 13 forest plots, training and test splits were merged, leaving a 75/25 split for training and internal validation, respectively. The model was then tested on the 25% test split of the withheld, unseen plot. This provides a direct comparison of the generalizability results to the main benchmark as consistent test-data splits are used.

2.3.5. Model training and configuration

All model developments in this study were performed using the Drago Supercomputer (<https://aic.csic.es/servicios/computacion/hpc-drago>), a high-performance computing (HPC) cluster managed by

the Spanish National Research Council (CSIC). Computational tasks were executed on an NVIDIA H200 GPU with 141 GB of VRAM; however, the VRAM was capped at 40 GB for all experiments. Additionally, a maximum training window of 24 h was established for each experiment. This threshold was used to identify models that, while potentially accurate, may be computationally impractical for large-scale forest experiments. Computations were executed using PyTorch 2.2.0 and CUDA 12.1.

We tested two configurations of PointNeXt: a smaller version (PointNeXt-S) and a larger one (PointNeXt-L), which includes InvResMLP blocks (Fig. 1) to improve performance at the cost of additional memory and computation (Qian et al., 2022). For SPT, we tested an additional in-house implementation (hereafter referred to as “SPT-Delaunay”) with improved portability and CPU pre-processing. In this version, the superpoint linkage is based on Geogram’s highly optimized triangulation method (Lévy & Filbois, 2015), which is the linkage method employed in the original SuperPoint Graph architecture (Landrieu and Simonovsky, 2018).

To evaluate the architectures in their standard forms, all models were tested using their reference open-source implementations (Qian et al., 2022; Pointcept Contributors, 2023; Robert et al., 2023) and default configurations, strictly avoiding model-specific hyperparameter tuning.

Additionally, to ensure broad contextual awareness, we maximized batch size and minimized the number of batches during training as much as possible for all models. This is particularly important for distinguishing overlapping classes like canopy and understorey. The main differences in model configuration are shown in Table 2 for reference. To ensure reproducibility, full training configurations, model weights, and settings are available in the Supplementary Material.

2.3.6. Evaluation metrics

To follow the standardized evaluation protocols provided by the original model repositories and to facilitate direct comparison with existing 3D benchmarks, performance was assessed using the following accuracy metrics: Overall Accuracy (OA), mean class Accuracy (mAcc), class-wise Intersection over Union (IoU), and mean Intersection over Union (mIoU). These metrics are widely used to assess 3D point cloud segmentation performance (Kulicki et al., 2024; Sarker et al., 2024). OA is defined as the ratio of correctly predicted points to the total number of points:

$$OA = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP are true positive points, TN are true negative points, FP are false positive points and FN are the false negative points. mAcc is the average of the accuracy computed per each class:

$$mAcc = \frac{1}{C} \sum_{c=1}^C Acc_c, Acc_c = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c},$$

where C is the total number of classes and Acc_i is the accuracy calculated for the i -th class.

Class-wise IoU measures the overlap (intersection) between the predicted points of a specific class and the ground-truth points of that class, divided by the total area covered by both (union):

$$IoU_c = \frac{Intersection}{Union} = \frac{TP_c}{TP_c + FP_c + FN_c}$$

Intersection is the set of points that are correctly predicted as belonging to a specific class. This corresponds to the True Positives for the class c (TP_c). *Union* is the total number of points in both the predicted and actual labels for the class, which includes TP_c , (points correctly pre-

Table 2

Main configuration differences of each architecture. Batch size of 1 means the whole point cloud is passed to the network as input. “PointNeXt” refers to both PointNeXt-L and PointNeXt-S and “SPT” applies to both SPT and SPT-Delaunay models. To ensure reproducibility, full training configurations, model weights and settings are available in the Supplementary Material.

Configuration	PointNeXt	SPT	PTV3	OA-CNN
Scheduler	Cosine Annealing	Cosine + Warmup	OneCycleLR	OneCycleLR
Epochs	300 (3 x 100)	450	300	300
Loss Function	CrossEntropy	CrossEntropy + Kullback-Leibler	CrossEntropy + Lovász-SoftMax	CrossEntropy
Batch Size	1	4	1	1
Data Augmentation	Rotation, Scale, Jitter	Tilt/Rotation, Anisotropic Scale	Rotation, Scale, Jitter, Flip	Rotation, Scale, Jitter, Flip

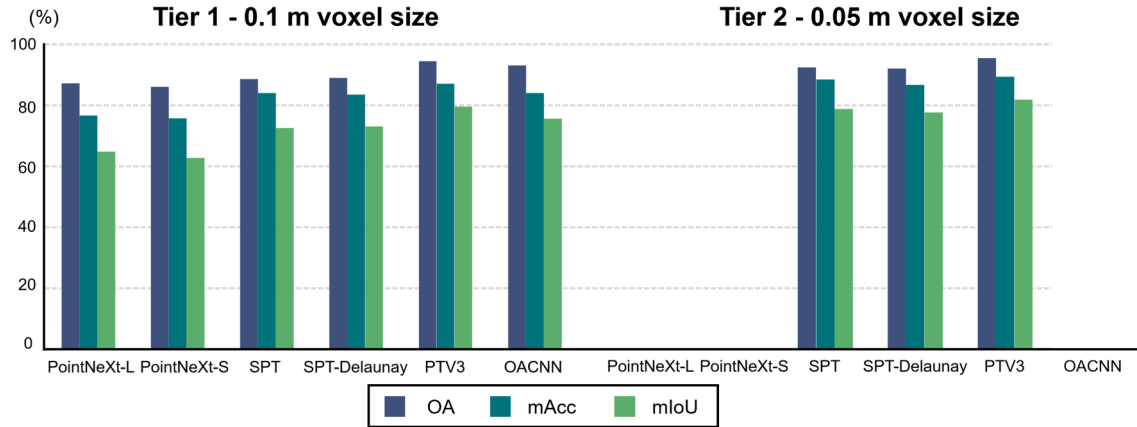


Fig. 5. Accuracy metrics (OA, mAcc, mIoU) of all benchmarked models across all plots. Left: Results from Tier 1 benchmark (using a voxel size of 0.1 m). Right: Results from Tier 2 benchmark (using a voxel size of 0.05 m): PointNeXt-L exceeded VRAM limit, PointNeXt-S exceeded time limit and OA-CNN ran into integer overflow in this benchmark tier.

dicted), FP_c (points incorrectly predicted), and FN_c , points of class c that were missed by the model. Lastly, the mIoU is the average IoU across all semantic classes. It is computed in a similar way to mAcc, but averaging class-specific IoU (IoU_c) instead of class-specific accuracy:

$$mIoU = \frac{1}{C} \sum_{c=1}^C IoU_c$$

All metrics were computed on the test sets using the predicted labels mapped back to the original resolution via voxel-label remapping. Furthermore, to support extended analyses, we provide the full set of point-wise predictions in the Supplementary Materials.

Training and inference runtimes were registered as well. We registered training runtime as the sum of pre-processing (any operation done by the models to digest the point clouds), training and testing runtime averaged across all plots. The inference runtime was calculated as the time required to perform inference in the whole Plot 1 (not just the test split), averaged across 5 runs. This provides a reference value of what time it takes for each model to perform inference on a more realistic scenario.

3. Results

In this section we evaluate the performance of the models for semantic segmentation of forest point clouds. All the results are computed in the original point cloud resolution, after voxel-to-point label remapping. We structure the evaluation in four parts. First, we provide general benchmark results where we compare overall performance across models and across tiers (Section 3.1, Figs. 5 and 6, Tables 3 and 4). Second, we assess how different forest structural configurations affect model performance (Section 3.2, Figs. 7 and 8, Tables 5–8). Third, we analyse the generalization capability of the models in unseen scenes (Section 3.3, Tables 9 and 10). Finally, we examine trade-offs in ease of

implementation and runtime (Section 3.4, Table 11). All code, raw accuracy results and predicted labels are available in Supplementary Materials for reproducibility.

3.1. Model comparison

3.1.1. Tier 1 benchmark (0.1 m voxel size)

The Tier 1 benchmark established a level playing field to evaluate architectural performance under identical constraints (Figs. 5 and 6, Table 3).

PTV3 clearly outperformed all other architectures, achieving the highest scores across all global metrics and leading in every class-wise IoU. OA-CNN emerged as the second-most performant model. The SPT and SPT-Delaunay models showed moderate consistency, while PointNeXt-L yielded the lowest overall scores. The semantic separation of the understorey was particularly struggling for every model.

3.1.2. Tier 2 benchmark (0.05 m voxel size)

The Tier 2 benchmark (Figs. 6 and 7, Table 4) served as a scalability stress-test, revealing clear differences in the computational efficiency of the architectures. Several models could not handle the increased point density and memory overhead at this resolution: PointNeXt-L and PointNeXt-S were excluded after exceeding the VRAM and training-time thresholds, respectively, while OA-CNN was limited by the point-processing capacity of its underlying spconv dependency (SPConv Contributors, 2022), resulting in an integer-overflow error.

For the “surviving” architectures (PTV3, SPT, and SPT-Delaunay), the transition to 0.05 m resolution yielded consistent performance improvements. PTV3 maintained its lead, reaching a peak mIoU of 81.9% and demonstrating superior accuracy in resolving canopy and stems. SPT variants showed the most significant relative gains in this tier, with mIoU increasing by over 6 percentage points compared to tier 1. This places them ahead of OA-CNN in terms of segmentation accuracy.

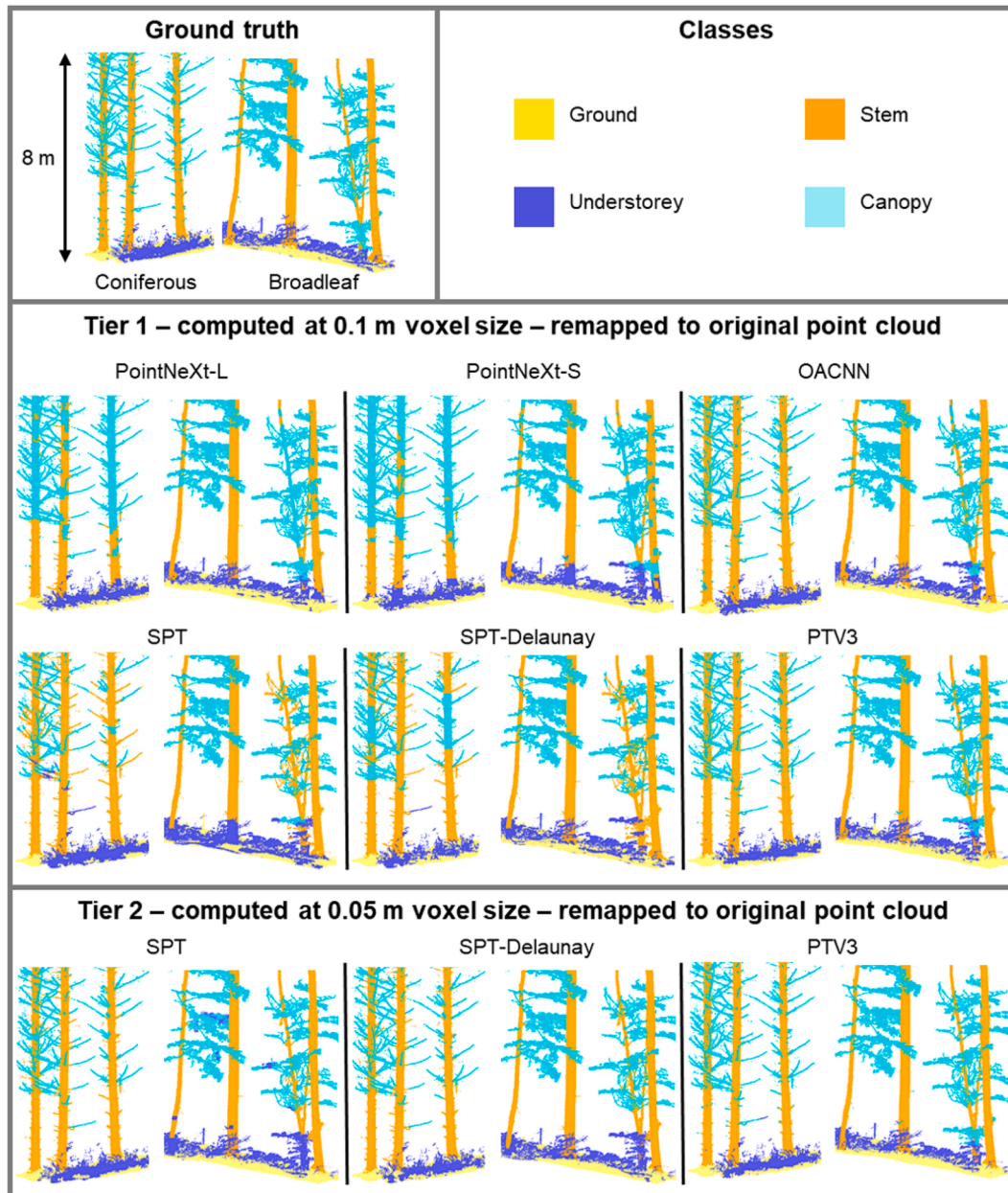


Fig. 6. Visualization of the inference in the test dataset in the original point density. As examples, a broadleaf (Plot 6) and a coniferous (Plot 7) plot are shown. A crop of the test point clouds is shown; the complete results are available in Supplementary Materials.

Table 3

Inference results on the test splits for Tier 1 benchmark (0.1 m voxel size). Metrics are given as percentages (%), OA: Overall Accuracy; mAcc: mean class Accuracy; mIoU: mean Intersection over Union; IoU: Class-specific Intersection over Union. Bold values indicate the highest value per each accuracy metric.

Model	OA	mAcc	mIoU	Understorey IoU	Ground IoU	Stem IoU	Canopy IoU
PointNeXt-L	87.1	76.6	64.7	47.5	75.7	50.2	85.5
PointNeXt-S	86.0	75.7	62.8	45.6	76.0	44.6	85.0
PTV3	94.5	87.1	79.5	56.4	80.9	84.9	95.7
SPT	88.5	84.0	72.6	44.1	73.0	77.2	90.5
SPT-Delaunay	89.0	83.4	73.0	43.9	77.6	74.4	90.6
OA-CNN	93.0	84.0	75.6	52.0	75.1	80.9	94.4

Table 4

Inference results on the test splits for Tier 2 benchmark (0.05 m voxel size). Metrics are given as percentages (%), OA: Overall Accuracy; mAcc: mean class Accuracy; mIoU: mean Intersection over Union; IoU: Class-specific Intersection over Union. Bold values indicate the highest value per each accuracy metric.

Model	OA	mAcc	mIoU	Understorey IoU	Ground IoU	Stem IoU	Canopy IoU
PointNeXt-L				Exceeded VRAM limit			
PointNeXt-S				Exceeded time limit			
PTV3	95.5	89.4	81.9	58.4	83.0	89.2	96.9
SPT	92.4	88.5	78.7	52.6	80.4	83.7	93.3
SPT-Delaunay	92.0	86.6	77.6	50.8	78.6	82.9	93.1
OA-CNN				Hit integer overflow			

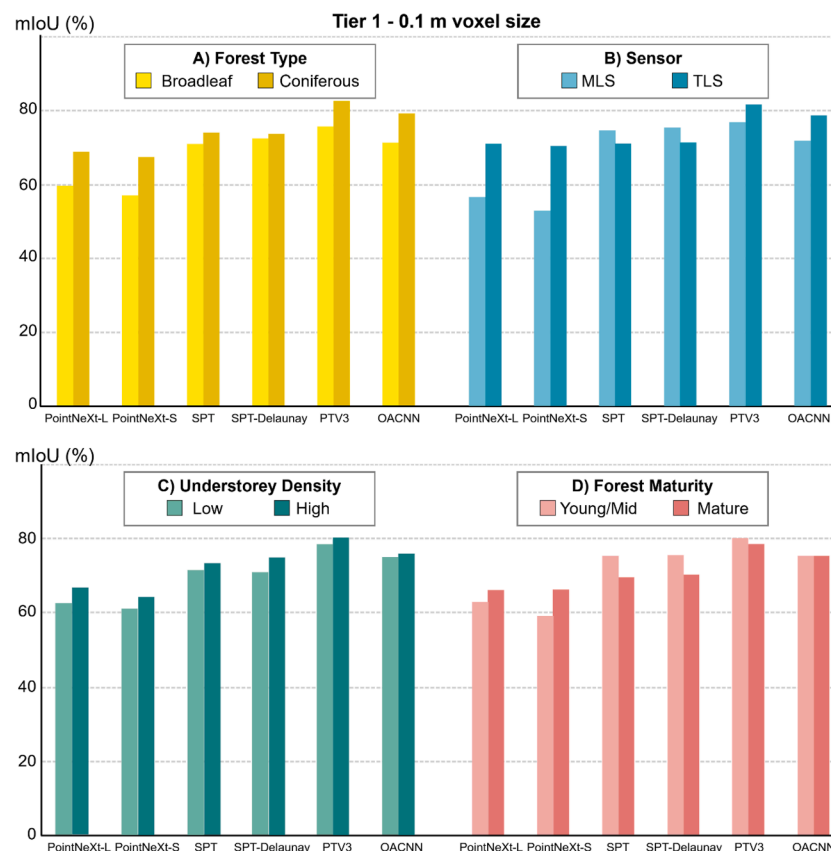


Fig. 7. Tier 1 (0.1 m voxel size) mIoU of all benchmarked models disaggregated by data partitions. A) Results disaggregated by forest type. B) Results disaggregated by Sensor. C) Results disaggregated by Understorey Density. D) Results disaggregated by Forest Maturity.

3.2. Semantic segmentation by structure

In this section we explore how the segmentation results vary when they are disaggregated by the data partitions introduced in 2.2.1. Fig. 7 visually displays the disaggregated results of Tier 1 of the benchmark and Fig. 8 shows those of Tier 2.

3.2.1. Forest type

When disaggregating results by forest type (Figs. 7 and 8, Table 5),

coniferous plots always yielded higher mIoU values than broadleaf plots. On average, coniferous plots yielded an mIoU increase of 6.4% in Tier 1 and 5.3% in Tier 2 compared to broadleaf stands.

The largest per-class disparity was found in the Understorey IoU, where coniferous plots outperformed broadleaf plots by an average of 21.2% in Tier 1 and 24.5% in Tier 2 (Fig. 7). Models like SPT and SPT-Delaunay saw their understorey segmentation nearly double in accuracy when transitioning from broadleaf to coniferous environments. While the majority of classes favoured coniferous structures, the Ground class

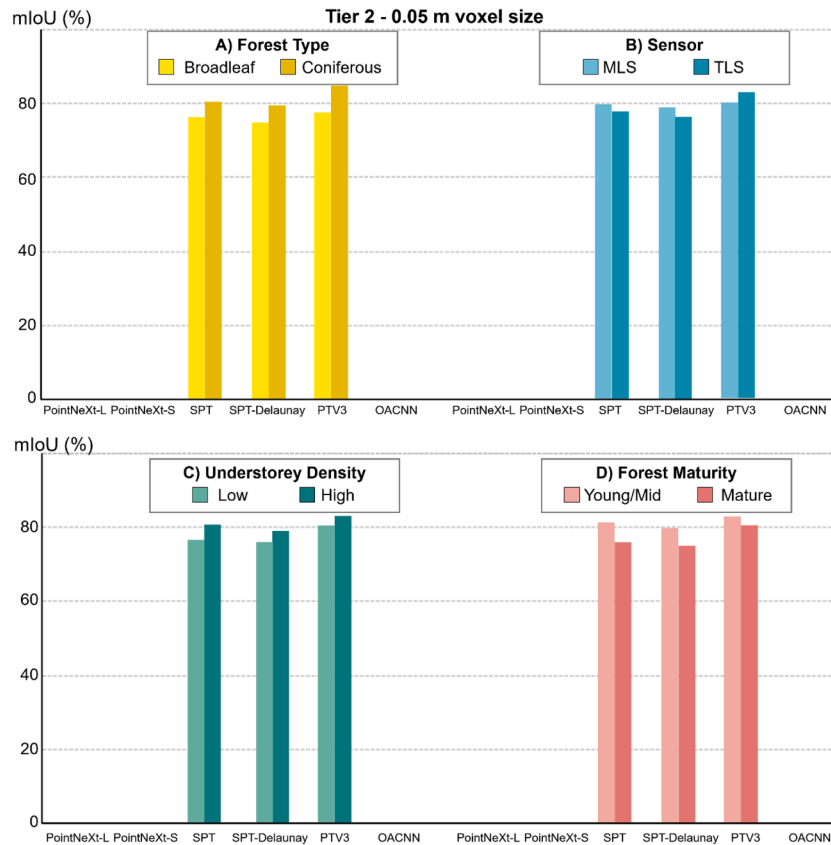


Fig. 8. Tier 2 (0.05 m voxel size) mIoU of all benchmarked models disaggregated by data partitions. A) Results disaggregated by forest type. B) Results disaggregated by Sensor. C) Results disaggregated by Understorey Density. D) Results disaggregated by Forest Maturity. PointNeXt-L exceeded VRAM limit, PointNeXt-S exceeded time limit and OA-CNN ran into integer overflow in this benchmark tier.

Table 5

Inference results aggregated by forest type. Accuracy metrics are given as percentages (%). Average difference between coniferous and broadleaf plots is shown. Positive values mean higher segmentation accuracy in coniferous plots. Bold values indicate the highest value per each accuracy metric.

Model	ForestType	OA		mAcc		mIoU		Understorey IoU		Ground IoU		Canopy IoU		Stem IoU	
		Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2
PointNeXt-L	Broadleaf	86.2	–	74.9	–	59.5	–	35.0	–	72.6	–	85.4	–	45.2	–
	Coniferous	87.9	–	77.8	–	68.6	–	56.8	–	78.1	–	85.7	–	54.0	–
PointNeXt-S	Broadleaf	84.6	–	73.6	–	56.8	–	33.7	–	70.9	–	84.3	–	38.5	–
	Coniferous	87.0	–	77.2	–	67.2	–	54.5	–	79.8	–	85.5	–	49.2	–
SPT	Broadleaf	88.1	91.4	82.4	88.6	70.8	76.4	31.4	38.4	69.5	79.6	90.8	92.4	78.8	83.4
	Coniferous	88.9	93.1	85.2	88.4	73.9	80.5	53.7	63.2	75.6	81.0	90.3	94.0	76.0	84.0
SPT-Delaunay	Broadleaf	89.1	90.8	82.6	85.6	72.4	75.0	32.8	35.3	75.1	78.7	90.9	91.9	78.0	82.5
	Coniferous	88.9	92.8	84.1	87.4	73.4	79.6	52.2	62.4	79.4	78.5	90.4	94.0	71.7	83.3
PTV3	Broadleaf	93.9	95.0	85.5	88.3	75.4	77.8	44.4	46.0	80.5	82.1	94.4	95.8	82.4	87.1
	Coniferous	94.9	95.9	88.3	90.2	82.5	85.0	65.5	67.6	81.1	83.7	96.7	97.7	86.8	90.8
OA-CNN	Broadleaf	92.2	–	82.0	–	71.1	–	39.6	–	73.5	–	92.9	–	78.5	–
	Coniferous	93.6	–	85.6	–	79.0	–	61.4	–	76.3	–	95.5	–	82.6	–
Average difference:		+1.18	+1.5	+2.87	+1.2	+6.43	+5.3	+21.2	+24.5	+4.7	+1.0	+0.9	+1.8	+3.15	+1.7

showed the smallest margin of difference (+4.7% in Tier 1).

In the Tier 2 benchmark (0.05 m voxel size), the performance gap between forest types remained stable (Fig. 8). Although the finer resolution improved absolute scores for both types, it did not mitigate the inherent difficulty of broadleaf scenes. For the PTV3 model, coniferous plots reached a peak Canopy IoU of 97.7% and a Stem IoU of 90.8%, representing the highest class-wise accuracies recorded in this study.

3.2.2. Sensor

The results grouped by sensor type (Figs. 7 and 8, Table 6) show that TLS generally provides a more favourable environment for semantic segmentation, though this effect is model-dependent. In the Tier 1

benchmark (Fig. 7), TLS led to an average increase in mIoU of 6.0%. This gap was most extreme for the PointNeXt architectures, which saw mIoU improvements of over 14% when transitioning from MLS to TLS.

However, the SPT and SPT-Delaunay models broke this trend, demonstrating superior performance in MLS acquisitions (e.g., SPT Tier 1 mIoU: 74.6% in MLS vs. 71.0% in TLS). While TLS provided a massive boost for the Ground (+14.8%) and Stem (+13.9%) classes, the Understorey IoU was consistently higher in MLS setups across all models and both tiers (−4.5% in Tier 1 and −5.7% in Tier 2). In Tier 2 (Fig. 8, Table 6), the performance gap virtually disappeared for global metrics due to SPT variants performing better on MLS data and PTV3 performing better on TLS data. The class-specific divergence persisted, as TLS

Table 6
Inference results aggregated by Sensor. Accuracy metrics are given as percentages (%). Average difference between MLS and TLS plots is shown. Positive values mean higher segmentation accuracy in TLS set ups. Bold values indicate the highest value per each accuracy metric.

Model	Sensor	OA		mAcc		mIoU		Understorey IoU		Ground IoU		Canopy IoU		Stem IoU	
		Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2
PointNeXt-L	MLS	82.5	–	71.8	–	56.5	–	48.4	–	63.7	–	81.5	–	32.5	–
	TLS	90.7	–	80.2	–	70.9	–	46.8	–	84.7	–	88.6	–	63.5	–
PointNeXt-S	MLS	80.0	–	69.3	–	52.8	–	46.6	–	63.3	–	79.3	–	21.8	–
	TLS	90.5	–	80.4	–	70.3	–	44.8	–	85.5	–	89.2	–	61.7	–
SPT	MLS	89.3	92.1	84.0	91.5	74.6	79.9	48.9	55.5	68.3	76.4	90.9	93.1	77.5	82.7
	TLS	88.0	92.6	84.0	86.2	71.0	77.9	40.5	50.4	76.5	83.4	90.2	93.4	76.9	84.5
SPT-Delaunay	MLS	89.6	91.7	84.6	89.4	75.3	79.0	48.8	54.1	72.3	76.4	91.1	92.6	76.4	81.6
	TLS	88.5	92.2	82.5	84.6	71.2	76.5	40.3	48.3	81.5	80.2	90.3	93.5	72.9	84.0
PTV3	MLS	92.9	94.5	86.7	89.9	76.9	80.2	58.8	61.9	74.7	77.3	93.6	95.5	80.3	86.2
	TLS	95.6	96.2	87.3	89.0	81.5	83.1	54.7	55.7	85.5	87.3	97.4	97.9	88.4	91.5
OA-CNN	MLS	91.0	–	82.0	–	71.6	–	53.5	–	65.1	–	92.0	–	75.9	–
	TLS	94.5	–	85.6	–	78.6	–	50.9	–	82.6	–	96.2	–	84.6	–
Average difference:		+3.7	+0.9	+3.6	−3.7	+6.0	−0.6	−4.5	−5.7	+14.8	+6.9	+3.9	+1.2	+13.9	+3.1

Table 7
Inference results aggregated by Understorey Density. Accuracy metrics are given as percentages (%). Average difference between low-density and high-density plots is shown. Positive values mean higher segmentation accuracy in high-density understorey. Bold values indicate the highest value per each accuracy metric.

Model	Understorey Density	OA		mAcc		mIoU		Understorey IoU		Ground IoU		Canopy IoU		Stem IoU	
		Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2
PointNeXt-L	Low	87.5	–	76.5	–	62.6	–	32.4	–	87.4	–	83.4	–	47.3	–
	High	86.8	–	76.7	–	66.9	–	62.5	–	64.1	–	87.7	–	53.2	–
PointNeXt-S	Low	87.0	–	76.4	–	61.2	–	29.9	–	89.0	–	83.8	–	42.1	–
	High	85.0	–	74.9	–	64.4	–	61.2	–	63.0	–	86.1	–	47.1	–
SPT	Low	90.9	93.7	84.1	86.9	71.7	76.7	29.7	38.0	85.7	89.1	92.4	94.9	78.9	84.6
	High	86.2	91.1	83.9	90.2	73.5	80.8	58.5	67.1	60.3	71.6	88.6	91.7	75.4	82.9
SPT-Delaunay	Low	90.9	93.6	82.2	84.8	71.0	76.0	30.3	36.4	87.7	88.9	91.3	94.8	74.7	84.0
	High	87.1	90.3	84.7	88.5	75.0	79.1	57.6	65.2	67.5	68.3	89.9	91.4	74.1	81.8
PTV3	Low	95.7	96.6	86.5	89.0	78.6	80.6	42.5	44.3	92.2	92.6	95.5	96.8	84.1	88.4
	High	93.2	94.4	87.7	89.8	80.4	83.2	70.4	72.4	69.6	73.4	95.9	96.9	85.8	90.0
OA-CNN	Low	94.2	–	84.1	–	75.1	–	37.7	–	89.7	–	93.8	–	79.1	–
	High	91.8	–	84.0	–	76.1	–	66.4	–	60.6	–	94.9	–	82.6	–
Average difference:		−2.7	−2.7	+0.4	+2.6	+2.7	+3.3	+29.0	+28.7	−24.4	−19.1	+0.5	−2.2	+0.5	−0.8

Table 8
Inference results aggregated by Forest Maturity. Accuracy metrics are given as percentages (%). Average difference between young/mid and mature plots is shown. Positive values mean higher segmentation accuracy in mature plots. Bold values indicate the highest value per each accuracy metric.

Model	Forest Maturity	OA		mAcc		mIoU		Understorey IoU		Ground IoU		Canopy IoU		Stem IoU	
		Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2	Tier 1	Tier 2
PointNeXt-L	Young/Mid	83.9	–	76.5	–	63.0	–	48.9	–	73.7	–	79.4	–	50.2	–
	Mature	90.4	–	76.7	–	66.4	–	46.1	–	77.8	–	91.7	–	50.2	–
PointNeXt-S	Young/Mid	81.4	–	74.8	–	59.2	–	45.4	–	73.1	–	77.3	–	41.1	–
	Mature	90.6	–	76.6	–	66.3	–	45.7	–	78.9	–	92.7	–	48.1	–
SPT	Young/Mid	89.1	92.5	84.7	90.3	75.5	81.4	47.2	55.3	73.4	81.7	89.1	92.1	81.4	86.6
	Mature	87.9	92.3	83.3	86.7	69.6	76.0	41.1	49.8	72.6	79.1	91.9	94.5	73.0	80.8
SPT-Delaunay	Young/Mid	89.3	91.8	83.7	88.8	75.7	80.0	47.1	52.6	76.8	80.8	88.4	91.4	79.5	85.3
	Mature	88.7	92.2	83.2	84.5	70.3	75.2	40.8	49.0	78.4	76.4	92.8	94.8	69.3	80.6
PTV3	Young/Mid	94.1	95.5	87.5	89.9	80.3	83.0	59.9	62.2	80.7	83.1	93.9	95.8	86.6	91.1
	Mature	94.8	95.4	86.7	88.9	78.7	80.7	53.0	54.5	81.1	83.0	97.5	97.9	83.2	87.4
OA-CNN	Young/Mid	92.3	–	84.2	–	75.5	–	54.3	–	73.3	–	91.9	–	82.7	–
	Mature	93.7	–	83.9	–	75.7	–	49.8	–	76.9	–	96.9	–	79.0	–
Average difference:		+2.6	0.0	−0.2	−2.9	−0.4	−4.2	−4.4	−5.6	+2.5	−2.4	+7.3	+2.7	−3.1	−4.7

Table 9
Generalization study results using a 0.1 m voxel size. CT subscript means the metrics are averaged across all 14 folds of the LOO-CT. Bold values indicate the highest value per each accuracy metric.

Model	OA _{CT}	mAcc _{CT}	mIoU _{CT}	Understorey IoU _{CT}	Ground IoU _{CT}	Stem IoU _{CT}	Canopy IoU _{CT}
PTV3	92.9	85.9	78.8	58.0	79.5	83.9	93.9
SPT	90.2	81.5	71.5	46.6	73.9	75.6	89.8
SPT-Delaunay	89.5	81.4	70.1	44.2	75.2	72.2	88.7

Table 10

Generalization study results using a 0.05 m voxel size. CT subscript means the metrics are averaged across all 14 folds of the LOO-CT. Bold values indicate the highest value per each accuracy metric.

Model	OA _{CT}	mAcc _{CT}	mIoU _{CT}	Understorey IoU _{CT}	Ground IoU _{CT}	Stem IoU _{CT}	Canopy IoU _{CT}
PTV3	93.4	87.1	80.4	60.0	79.0	88.0	94.5
SPT	92.9	87.7	77.7	58.2	80.2	81.1	91.4
SPT-Delaunay	92.8	86.3	77.9	58.0	81.7	80.6	91.2

Table 11

Training and inference runtimes of all model variations. Inference runtime is measured as the time required to perform inference in the complete Plot 01, averaged across five repetitions. * Inference times exclude the time required to compute Z₀ using 3DFin, which is approximately 58 s.

Model	Voxel size (m)	TrainingRuntime (h)	InferenceRuntime (s)*
PointNeXt-L	0.1	6.3	57
PointNeXt-L	0.05	Exceeded VRAM limit	
PointNeXt-S	0.1	5.9	55
PointNeXt-S	0.05	Exceeded time limit	
SPT	0.1	0.7	14
	0.05	1.0	23
SPT-Delaunay	0.1	0.6	9
	0.05	0.9	12
PTV3	0.1	0.6	4
	0.05	1.3	16
OA-CNN	0.1	0.8	6
OA-CNN	0.05	Integer overflow	

continued to favour stem and ground resolution, while MLS provided an advantage for understorey and canopy delineation.

3.2.3. Understorey density

The disaggregation of results by Understorey Density (Figs. 7 and 8, Table 7) reveals a divergence between semantic accuracy and global geometric consistency. Across all models and both tiers, mIoU was consistently higher in high-density plots (+2.7% in Tier 1; +3.3% in Tier 2), despite a general decrease in OA (−2.7%). However, this gain came at the expense of the Ground class, which suffered a sharp average decline of 24.4% in Tier 1 (Fig. 7). In the Tier 2 benchmark (Fig. 8), the higher resolution mitigated the ground-degradation effect slightly for SPT models but reinforced the understorey segmentation gains. In Tier 2, the high-density understorey plots yielded the highest mAcc scores of the entire experiment (reaching 90.2% for SPT), suggesting that a denser understorey provides a more recognizable geometric signature for deep learning architectures, even as it complicates the delineation of the terrain below.

3.2.4. Forest Maturity

Disaggregating the results by forest maturity (Figs. 7 and 8, Table 8) reveals a complex relationship between stand age and segmentation accuracy. On average, mature plots yielded a higher OA (+2.6% in Tier 1), yet showed a decline in mIoU (particularly in the Tier 2 benchmark), where mIoU dropped by an average of 4.2% compared to younger stands. The most notable class-wise trend was the significant increase in Canopy IoU, which rose by an average of 7.3% in Tier 1 (Fig. 7). This gain was especially pronounced in the PointNeXt architectures, where Canopy IoU jumped by over 12% in mature plots. Conversely, Understorey and Stem IoU scores generally declined in mature environments (−4.4% and −3.1% in Tier 1, respectively). Interestingly, while the higher-performing models like PTV3 and SPT versions saw their mIoU scores decrease when moving from young to mature stands, the PointNeXt models exhibited the opposite behaviour, achieving higher mIoU scores in mature environments (Figs. 7 and 8).

3.3. Model generalizability

To assess the spatial generalizability of the results, we implemented a Leave-One-Plot-Out Cross-Testing (LOO-CT) protocol. The results from this analysis (averaged across the 14 folds) are shown in Tables 9 and 10. OA-CNN and PointNeXt were excluded from the generalizability study due to the same technical and scalability bottlenecks that arose in the Tier 2 benchmark. Specifically, OA-CNN encountered the same integer overflow issues when handling the increase training data volume, and PointNeXt exhibited significantly lower training efficiency, making a 14-fold cross-testing cycle computationally prohibitive within the standardized HPC window even at the 0.1 m resolution. In this regard, SPT, SPT-Delaunay and PTV3 implementations demonstrated the highest computational efficiency and architectural stability again, considering the high computational demand of performing 14 independent training and testing cycles.

At the 0.1 m voxel size, a slight performance decay was observed when predicting on entirely unseen plots (Table 9). PTV3 emerged as the most robust architecture, maintaining a 78.8% mIoU, a marginal 0.7% decrease from the standard 50/25/25 split. For the SPT family, the mIoU remained highly consistent, with only a marginal 1.1% decrease for the base SPT and a 2.9% decrease for SPT-Delaunay. The primary source of this decay was the Understorey class, where IoU dropped significantly compared to the standard benchmark. Notably, the Overall Accuracy (OA) for both SPT models increased during cross-testing (SPT: +1.7%), suggesting that the additional training data in the LOO-CT protocol helped the models better handle the global geometry of unseen plots.

At the higher resolution (0.05 m voxel size), the generalizability of the models improved (Table 10), narrowing the gap between the standard benchmark (Table 4) and the LOO-CT results. In this regard, the SPT models showed resilience: SPT-Delaunay improved its mIoU from 77.6% (Table 4) to 77.9% (Table 10). This trend suggests that higher spatial resolution is beneficial for spatial generalizability, as finer resolutions help to resolve the more complex branching and foliage patterns that define the forest architecture. Moreover, SPT-Delaunay's Understorey IoU jumped from 50.8% (Table 3) to 58.0% (+7.2%) (Table 10). This could indicate that the increased training volume in the LOO protocol was particularly beneficial for the Understorey class, which represents the most geometrically variable (and possibly “data-hungry”) component of the dataset.

3.4. Runtimes and computational efficiency

To assess runtime and computational efficiency, training runtime (the time required to complete the training, evaluation and testing of each model in the general benchmark) and inference runtime (the time required to perform inference on Plot 1) were recorded (Table 11). Evaluating these metrics reveals trade-offs between segmentation accuracy and computational efficiency. PointNeXt models emerged as the most computationally demanding, requiring over 6 h for training and nearly 1 min for inference per plot at 0.1 m resolution. Given that these models also yielded the lowest segmentation scores, their high computational cost suggests a lack of suitability for large-scale forest studies compared to more modern alternatives.

Notably, at the 0.1 m scale, PTV3 offered the fastest inference time (4 s) and tied with SPT-Delaunay for the fastest training time (0.6 h), while still outperforming every other architecture in accuracy (Tables 3

and 4) (even those trained at the finer 0.05 m resolution). OA-CNN also showed strong training efficiency (0.8 h) and competitive inference (6 s), though it remained constrained by its underlying software dependencies at the higher resolution. The most notable scalability trend was observed within the SPT family. Moving from the 0.1 m to the 0.05 m resolution resulted in a training runtime increase of only 0.3 h (from 0.7 h to 1.0 h in base SPT and from 0.6 h to 0.9 h in SPT-Delaunay). Moreover, at the finer 0.05 m resolution, both SPT variants trained faster than PTV3, which required 1.3 h and SPT-Delaunay outperformed PTV3 in inference time.

4. Discussion

In this study, we performed an evaluation of recent deep learning models for the semantic segmentation of ground-based LiDAR point clouds in forest environments. Among all tested architectures, transformer-based models performed the best. Particularly, PTV3 achieved the highest overall performance, reaching an mIoU of 81.9%, OA of 95.5%, and mAcc of 89.4%. This model consistently outperformed others across all semantic classes and demonstrated generalizability, maintaining stable performance levels when evaluated on entirely unseen forest plots in the LOO-CT protocol. Architecturally, this superiority could be attributed to its self-attention mechanisms and serialization strategy (Wu et al., 2024), which effectively capture long-range geometric dependencies across the forest scene, allowing for better discrimination of complex, overlapping structures. SPT showed the second-best accuracy but demonstrated the best training scalability, outpacing PTV3 in training runtime at the finer 0.05 m resolution. The abstraction of raw points into a coarser geometric graph allows SPT to efficiently “absorb” varying point densities during pre-processing, although this secondary quantization might limit its absolute point-level precision compared to other methods. OA-CNN achieved moderate results with highly competitive runtimes at the 0.1 m scale, but its performance remained below that of transformers and suffered from dependency issues. PointNeXt, however, segmented the point clouds with less accuracy while taking the longest runtime. This struggle likely comes from relying on iterative Farthest Point Sampling (FPS) to process the overly complex structures found in forest plots, which shows that the solutions found in more modern paradigms are noticeably more efficient for this task.

4.1. Effects of structural variability in segmentation

Our experiments revealed a clear and consistent trend: coniferous plots were easier to segment than broadleaf-dominated ones across nearly all metrics. When using the best-performing model (PTV3), coniferous plots achieved an average mIoU of 82.5%, compared to 75.4% in broadleaf plots. Similar patterns were observed for OA and mAcc. The largest differences were found in understory segmentation, where coniferous stands outperformed broadleaf by 21.2% in IoU. Stems and canopy segmentation also improved in coniferous plots, with gains of 4.4% and 2.3% in IoU, respectively. Coniferous stands in the dataset feature uniform, vertically aligned stems and more homogeneous canopy shapes, which facilitate clean separation between structural classes. In contrast, the broadleaf plots often contain irregular branching patterns, horizontal occlusions, and multi-layered understory components that make semantic boundaries harder to learn from 3D point clouds.

Segmentation accuracy also varied depending on the acquisition technology. Across all base models, TLS generally outperformed MLS, averaging an mIoU 6.0% higher. PTV3 reached an mIoU of 81.5% in TLS plots. Class-wise, the largest differences were obtained in ground and stems, where IoU differences are of 10.8% and 8.1% respectively. This difference could be attributed to the higher point density and greater measurement precision of TLS data, particularly in vertical and horizontal surfaces. Although MLS can mitigate occlusion effects by scanning from multiple trajectories, its data are generally noisier. This factor

may make semantic boundaries harder to learn, whereas the higher fidelity of TLS could provide clearer geometric signals that translate into more accurate segmentation. However, a notable exception was found in the SPT family, achieving slightly higher scores in MLS scans. This suggests that graph-based Superpoint features may act as a natural filter for the sensor-induced “jitter” and non-uniform densities of mobile scanning. Additionally, the multi-angle perspective of MLS appears to favour understory segmentation, which was consistently more accurate in mobile scans than in static TLS setups. Nevertheless, disentangling the effect of acquisition method independently from forest structure remains a challenge and will require future datasets with matched plots acquired using both techniques under controlled conditions.

Beyond forest type and sensors, understory complexity emerged as a primary driver of semantic accuracy. A “saliency threshold” was observed: understory IoU increased by a substantial 29.0% on average on “High” Understorey Density plots. This suggests that sparse understory is often confused with ground noise, whereas denser undergrowth provides a distinct geometric signature that architectures can more easily delineate. However, this gain in understory accuracy creates a direct trade-off with ground-level resolution; as shrub density increases, laser occlusion depletes the ground-point signal, leading to a drop in Ground IoU. Consequently, the understory remains the primary classification bottleneck across all tested architectures. Overcoming this limitation will likely require the continued expansion of training datasets to capture a wider variety of understory morphologies. Furthermore, as deep learning architectures become even more computationally efficient, the ability to process point clouds at even finer spatial resolutions (e.g., below 0.05 m) could provide the geometric fidelity needed to disentangle sparse, low-lying vegetation from the underlying terrain. In relation to this, several fine-scale classes are merged in the dataset. This aggregation is particularly detrimental to the understory, which currently functions as a catch-all class for highly heterogeneous elements ranging from woody shrubs and saplings to tall grasses. As the availability of open data expands, future work should incorporate more granular classes or hierarchical label structures to disentangle these elements.

Finally, forest maturity also revealed an architectural divergence in model behaviour. While mature stands provide a clearer “canopy”, likely due to the dominance of large crowns (+7.3% Canopy IoU), they also present higher individual geometric complexity that hinders stem classification. In this regard, the observed decrease in Stem IoU in mature plots (−3.1%) could be attributed to the fact that, as trees age, first-order branches increase in diameter and develop bark textures that become geometrically indistinguishable from the main trunk. This causes models to “absorb” these branches into the stem class rather than resolving them as part of the canopy. While high-performing architectures like PTV3 and SPT performed best in younger, high-density stands with simpler tree hierarchies, point-set architectures like PointNeXt proved more robust to the massive structural features of older trees. These results could suggest that the lower tree density found in mature forests simplifies the global scene for some models, even as the complex branching of individual mature trees complicates the specific segmentation of stems and undergrowth.

4.2. Practical considerations

Beyond segmentation accuracy, practical implementation factors such as processing time, memory requirements, and ease of deployment are critical in determining the applicability of software. The transition to a 0.05 m voxel resolution in Tier 2 served as a significant computational stress test, revealing the scalability limits of several architectures. While PTV3 and the SPT family handled the increased density with manageable overhead, the PointNeXt models proved prohibitively computationally expensive, exceeding the operational limits of 24 h training time and 40 GB VRAM. Furthermore, while OA-CNN showed great potential, it was limited by the point-processing capacity of its underlying spconv

dependency (SPConv Contributors, 2022), which triggered an integer-overflow error at this resolution.

Another practical consideration concerns the inference strategy applied to voxelized point clouds. PointNeXt and PointCept frameworks offer an alternative evaluation procedure in which inference is repeated multiple times per voxel, based on the number of points contained in the most populated voxel. This approach ensures that every original point is evaluated at least once and can mitigate the mixing of multiple semantic classes within a single voxel, which could lead to higher accuracy. However, this procedure would incur substantial runtime and memory costs, making it impractical for large-scale or operational forestry workflows. For this reason, and to ensure consistency across all models, we adopted the simpler and more scalable remapping strategy in which each voxel receives a single predicted label that is propagated to all points it contains.

Similarly, this benchmark utilized the default hyperparameters and configuration settings for each architecture to ensure a controlled and unbiased comparison. While this 'level playing field' was necessary for an initial evaluation, it likely overlooks the maximum performance potential of each model. Future research should explore task-specific hyperparameter optimization tailored to the unique geometric properties of forest data. This includes tuning specialized data augmentation strategies (such as the tilting rotations and the anisotropic scaling in SPT, which potentially simulates different tree shapes from the same training sampling) or the refinement of loss functions (like Lovász-SoftMax loss in PTV3, which is specifically designed to optimize mIoU in datasets with extreme class imbalances such as forests). Furthermore, hyperparameter tuning could alleviate the performance gap observed between acquisition modalities. For instance, adjusting the spatial receptive fields or modifying point-sampling strategies could help models better accommodate the sensor noise typical of MLS, potentially achieving more consistent segmentation quality across both TLS and MLS data. Transitioning from standardized baselines to fine-tuned, forest-specific configurations represents a logical next step in moving these models from experimental benchmarks to high-precision operational tools.

Lastly, all models were trained and tested under supervised learning settings, relying on complete and manually labelled data. This is an inherent characteristic of supervised learning, since the segmentation accuracy depends on availability and quality of manually labelled data (Sarker et al., 2024). Given this, and the lack of labelled datasets of forest point clouds, semi-supervised, weakly supervised, or self-supervised learning approaches could represent promising avenues to reduce dependence on extensive manual labeling (Sarker et al., 2024), until more publicly available datasets emerge. In this regard, a particularly promising avenue is the use of synthetic and semi-synthetic datasets. With engines like HELIOS++ (Winiwarter et al., 2022), it is possible to generate labelled forest scenes that simulate various sensor noise patterns and occlusions (Liu et al., 2026). Integrating these synthetic environments in the training dataset could allow models to encounter a much wider range of tree allometries and understorey conditions.

4.3. Runtime trade-offs

The evaluation of total runtime revealed substantial differences between architectures. PTV3 offered the best balance between segmentation quality and computational efficiency, requiring just 0.6–1.3 h to be trained. Notably, its training runtime remained low even at the 0.05 m resolution, which could largely be attributable to its use of space-filling curves for data serialization. The mapping of 3D spatial relationships into a 1-dimensional sequence that preserves locality allows the model to process high-density volumes with minimal memory overhead and linear scaling. This efficiency, combined with its top-tier performance, positions it as the most practically viable model in our benchmark. OA-CNN was less accurate but completed training in under one hour,

making it an attractive option when inference speed is a key concern and highlights sparse convolutions as a highly promising and scalable alternative to more computationally intensive transformer-based architectures. SPT presented good computational costs under its default implementation. Yet, its inference runtime dropped a further $\sim 50\%$ after plugging a faster Delaunay triangulation algorithm for neighborhood linkage, while maintaining similar mIoU scores across tests. These results suggest that the choice of spatial data structures plays a determinant role in the scalability of forest segmentation. Moreover, the shift to a 0.05 m resolution enabled a significant performance leap for the SPT models, allowing them to overtake OA-CNN in absolute mIoU. This indicates that SuperPoint-based models are particularly sensitive to geometric fidelity; at the 0.1 m scale, distinct structural elements may be collapsed into a single Superpoint, obscuring the graph's features. In this regard, while a fixed voxel resolution was maintained across all models to ensure a standardized input baseline, it is important to acknowledge that 'spatial resolution' is not strictly equivalent across different architectural paradigms. Each framework applies unique internal quantization and sampling strategies that further modify the density of the data. For example, the SPT framework performs Superpoint oversegmentation on top of the initial voxelization. Superpoints actually preserve higher semantic purity than voxels at equal scale (Robert et al., 2023). But, in the benchmark setup, they represent a secondary level of quantization, meaning that the SPT models reason on a 'coarser' graph representation than the rest. Thus, when the input point cloud is voxelized, SPT might require a higher voxel resolution to achieve a strictly 'fair' comparison against models like PTV3 or OA-CNN. Conversely, PointNeXt utilizes farthest point sampling within its hierarchical layers to subsample the cloud, which also introduces a density mismatch compared to the rest of the models.

4.4. Comparison to previous studies

Previous studies have performed semantic segmentation on complete forest scenes, including ground and understorey, assigning labels to every point in the point cloud (Krisanski et al., 2021; Kajaluoto et al., 2022; Shen et al., 2022; Ma et al., 2023; Oviedo de la Fuente et al., 2024). These studies provide a relevant basis for comparison to our results, as their objectives and methodologies are aligned with ours. Shen et al. (2022) reported high mIoU values (82.1–87.2%) using a modified PointCNN with geometric features. However, their taxonomy included a mixed "other" class, excluding detailed understorey classification. In contrast, our experiments explicitly preserve understorey as a separate category (a component of particular ecological relevance) and achieve comparable or better results under more complex segmentation conditions. Ma et al. (2023) achieved an mIoU of 69.0% and mAcc of 91.0% using PointNet with normal features. Our trained models outperform them by a wide margin in mIoU (up to 81.9%) across a broader and unaugmented dataset, highlighting the capacity of newer architectures to generalize in more diverse environments. Kajaluoto et al. (2022) explored the use of raw laser scanner measurements such as echo number, range, and reflectance alongside custom fully convolutional neural networks for semantic segmentation of MLS forest point clouds. Their taxonomy closely resembles ours, with four structural categories: ground, understorey, stem, and foliage (although branches were grouped with trunks in the 'stem' category). Their best-performing model reached a notable mIoU of 80.1% and OA of 93.1%, and they compare these with a standard implementation of RandLA-Net (Hu et al., 2020) which yielded an mIoU of 80.6% and an OA of 92.9%. While their accuracies are comparable to those reported in our work, their dataset is significantly smaller (13.7 million points from two plots), and the test data came from a subset of one of them. In contrast, our study covers over 850 million points across 14 plots with varied structure and acquisition conditions.

Krisanski et al. (2021) implemented a custom configuration of PointNet++ and reported an mIoU of 78.7% and an excellent OA of

95.4%. However, their evaluation excluded large portions of the canopy and their class definitions did not separate understorey and canopy. Despite these simplifications, their mIoU remains slightly lower than our best-performing configuration. While their OA is high, our results demonstrate that modern transformer-based architectures can match this level of accuracy under more complete and structurally heterogeneous conditions, while discriminating challenging vegetation layers like canopy versus understorey. [Oviedo de la Fuente et al. \(2024\)](#) provides the most directly comparable results, as they use the same four structural classes and evaluate segmentation on full forest plot TLS data. Their best-performing model, DGCNN, achieved an mIoU of 79.4%. In our experiments, the PTV3 consistently outperformed this, reaching mIoU values above 81% and achieving higher class-wise IoUs, particularly for the stems (up to 88.7%) and canopy (up to 94.8%). Furthermore, our results were obtained on a larger and more diverse dataset that includes both TLS and MLS scans. This introduces greater structural diversity, point density, and acquisition variability, further validating the robustness of our trained models under more realistic and wider applicable field conditions. Collectively, these comparisons show that modern transformer-based models, when combined with heterogeneous training data and appropriate feature integration, not only match but in several cases surpass the best results reported in the literature, even under more challenging segmentation setups.

In addition, while prior studies exist which report high accuracy (71.2–97.5% OA) for leaf-versus-wood segmentation across various architectures ([Wu et al., 2020](#); [Han & Sánchez-Azofeifa, 2022](#); [Alvites et al., 2023](#); [Dai et al., 2023](#); [Jiang et al., 2023](#); [Bryson et al., 2024](#); [Lu et al., 2025](#)), these primarily operate at the individual tree level, after filtering out the ground and understorey layers. This preprocessing alters segmentation dynamics, making direct comparisons to our plot-level finding difficult. [Xi et al. \(2023\)](#) used a dataset consisting of entire forest plots rather than individual trees, but their focus was restricted to distinguishing between stem and coarse woody debris, where they achieve an mIoU of 82%.

Similarly, while tree instance segmentation remains a highly active research field (e.g., [Straker et al., 2023](#); [Henrich et al., 2024](#); [Wielgosz et al., 2024](#); [Xiang et al., 2024](#); [Burmeister et al., 2025](#); [Cherlet et al., 2026](#)), its differing objectives and metrics limit direct comparisons. However, a recent large-scale benchmark ([Cherlet et al., 2026](#)) reporting plot-level results of up to 93% F1-score and Recall, and 95% Precision concluded that segmentation quality varied significantly across different forest types, a finding that agrees with the variability observed in our disaggregated results.

Finally, recent studies have also explored semantic segmentation using airborne and UAV laser scanning (ULS), achieving competitive results (e.g., 93.6–97.6% OA, 75.7–85.2% mIoU; [Xiang et al., 2024](#); [Lu et al., 2025](#)) but representing a complementary perspective rather than a direct equivalent. ULS provides relatively dense top-down coverage whereas TLS and MLS directly capture sub-canopy structures like stems, understorey, and ground elements. Thus, both modalities support distinct research goals and require tailored modelling strategies.

4.5. Operational Outlook and deployment strategies

The high accuracy and computational efficiency demonstrated by architectures like PTV3 and SPT open a new range of downstream applications for precision forestry. By providing a reliable automated separation of forest strata, these models facilitate the extraction of key inventory metrics such as Diameter at Breast Height (DBH) from the stem class, vertical fuel continuity for fire risk modelling from the understorey-canopy relationship, and aboveground biomass estimations. Furthermore, the ability to isolate the 'Ground' class with relatively high fidelity at a 0.05 m resolution could facilitate the generation of high-resolution Digital Terrain Models (DTMs) even under dense canopy cover.

Software dependencies within the Pointcept framework restrict

execution to CUDA-enabled environments, which in turn prevents CPU-based inference. On the other hand, SPT is 100% portable to CPU. Moreover, SPT-Delaunay can achieve faster training and inference times than PTV3 while being easier to deploy on a standard PC thanks to its lighter weight and a reduced set of CUDA-only software dependencies. Overall, these trade-offs suggest that model selection should be guided not only by accuracy and runtime, but also by resource efficiency, flexibility, and ease of use. To facilitate the adoption of these tools, we propose two distinct implementation pathways based on user capacity:

- (i) **Direct Inference (Operational Level):** For forestry practitioners and managers, the pre-trained weights released with this study allow for 'out-of-the-box' application. A significant finding of our benchmark is the hardware accessibility of certain models; for instance, the SPT architecture allows for efficient inference on standard CPUs or entry-level GPUs. This facilitates access to deep learning, enabling users without specialized HPC infrastructure to process large-scale plot data on local workstations.
- (ii) **Model Adaptation (Research & Development Level):** For advanced users with access to new datasets or specific forest types not covered here, our framework supports Transfer Learning. Rather than training from scratch, which requires significant computational resources, researchers can potentially fine-tune our provided weights with a smaller set of locally labelled data. This drastically reduces the entry barrier for developing site-specific models.

Nevertheless, although we have released all the code, model weights and configuration files, running pre-trained models via Python scripting could still be a burden for many forestry practitioners. In the future, the inclusion of pre-trained models within user-friendly software could unlock new applications of deep learning for structural forest analysis.

5. Conclusions

This study establishes a benchmark of state-of-the-art 3D deep learning architectures for the semantic segmentation of ground-based forest point clouds. Our findings emphasize the importance of selecting model architectures and features that are well-aligned with the structural characteristics of the target forest type. Transformer-based architectures consistently achieved the highest performance, with PTV3 attaining the best scores while maintaining competitive processing times and high generalizability. SPT and OA-CNN demonstrated effective trade-offs between accuracy and runtime. These differences are, however, subject to practical implications in terms of portability and ease of deployment in operational settings.

Conifer-dominated plots yielded higher segmentation accuracy than broadleaf-dominated plots, particularly for the stem and understorey classes. This difference likely reflects structural regularities in conifer stands, such as axis-aligned geometries, which facilitate semantic discrimination.

Understorey vegetation emerged as a key factor and a classification bottleneck, interacting strongly with forest type, scanning technology and ground segmentation. Providing models with sufficient training data to further separate distinct sub-categories of understorey vegetation could create more consistent geometric targets, potentially easing the current classification bottleneck. Conversely, the development of thick, "trunk-like" branches in mature stands leads to reduced stem segmentation accuracy in favour of the canopy class.

TLS plots were generally segmented more accurately, possibly due to higher geometric precision, while graph-based architectures proved remarkably robust to the noise inherent in MLS data.

Our results also demonstrate the value of open and diverse benchmark datasets, such as *SegmentedForests*, in advancing reproducible research in forest remote sensing. Still, future benchmarks should expand the scope by incorporating additional forest types such as

tropical, boreal, or open-canopy systems. Furthermore, we anticipate that future work will benefit from extending this benchmark with additional acquisition modalities, integrating synthetic and semi-synthetic training data, and adopting self-supervised learning regimes to reduce the manual labelling bottleneck. Finally, the integration of these segmentation methods into user-friendly software remains a critical next step.

Ultimately, this study provides both a scientific baseline and a practical roadmap for deployment. By releasing our pre-trained weights and inference tools, we enable two distinct implementation pathways: direct inference for operational forest structure assessments and transfer learning for researchers seeking to adapt these state-of-the-art models to new ecological contexts.

Supplementary Material

Code for PTV3 and OA-CNN implementations is available in https://github.com/3DFin/Pointcept_3DFin. Code for PointNeXt implementations is available in https://github.com/3DFin/PointNeXt_3DFin. Code for SPT implementations is available in https://github.com/3DFin/superpoint_transformer_3DFin.

Benchmark results are available in: <https://doi.org/10.5281/zenodo.17776120>.

The full dataset and 3DFin configuration files listing the parameters used to compute Z_0 can be found in <https://zenodo.org/records/17396681> (Laino et al., 2025).

Funding

This work was supported by the UK-RI NERC project grant [NE/T001194/1]: ‘Advancing 3D Fuel Mapping for Wildfire Behaviour and Risk Mitigation Modelling’, the Spanish Knowledge Generation project grant [PID2021-126790NB-I00]: ‘Advancing carbon emission estimations from wildfires applying artificial intelligence to 3D terrestrial point clouds’, grant [PRE2022-104159] funded by MCIN/AEI/<https://doi.org/10.13039/501100011033> and FSE+, research grant ‘FIREPROs’ [IDE/2024/000780] funded by the Principality of Asturias Government (Spain) and the US DoD SERDP/ESTCP project [RC20-1346].

CRedit authorship contribution statement

Diego Laino: Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Carlos Cabo:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Romain Janvier:** Writing – review & editing, Software, Methodology, Investigation, Data curation. **Celestino Ordonez:** Writing – review & editing, Methodology, Conceptualization. **Rodolfo Bolanos:** Writing – review & editing, Visualization. **Stefan H. Doerr:** Writing – review & editing, Resources. **Andrew T. Hudak:** Writing – review & editing, Resources. **Cristina Santin:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Alonso-Rego, C., Arellano-Pérez, S., Cabo, C., Ordoñez, C., Álvarez-González, J.G., Díaz-Varela, R.A., Ruiz-González, A.D., 2020. Estimating fuel loads and structural characteristics of shrub communities by using terrestrial laser scanning. *Remote Sens. (Basel)* 12 (22), 1–21. <https://doi.org/10.3390/rs12223704>.
- Alvites, C., Santopuoli, G., Hollaus, M., Pfeifer, N., Maesano, M., Moresi, F.V., Marchetti, M., Lasserre, B., 2021. Terrestrial laser scanning for quantifying timber

- assortments from standing trees in a mixed and multi-layered mediterranean forest. *Remote Sens. (Basel)* 13 (21). <https://doi.org/10.3390/rs13214265>.
- Alvites, C., Maesano, M., Molina-Valero, J.A., Lasserre, B., Marchetti, M., Santopuoli, G., 2023. Discrimination of leaves in a multi-layered mediterranean forest through machine learning algorithms. *Remote Sens. (Basel)* 15 (18), 4450. <https://doi.org/10.3390/rs15184450>.
- Bello, S.A., Yu, S., Wang, C., Adam, J.M., Li, J., 2020. Review: deep learning on 3d point clouds. *Remote Sens. (Basel)* 12 (11), 1729. <https://doi.org/10.3390/rs12111729>.
- Bello, I., Fedus, W., Du, X., Cubuk, E. D., Srinivas, A., Lin, T.-Y., Shlens, J., & Zoph, B. (2021). Revisiting ResNets: Improved training and scaling strategies. In *Advances in Neural Information Processing Systems* (Vol. 34). NeurIPS.
- Bryson, M., Ravendran, A., Mercier, C., Frickey, T., Jayathunga, S., Pearce, G., Hartley, R. J.L., 2024. Domain adaptation of deep neural networks for tree part segmentation using synthetic forest trees. *ISPRS Open Journal of Photogrammetry and Remote Sensing* 14, 100078. <https://doi.org/10.1016/j.ojphoto.2024.100078>.
- Buongiorno, J., Dahir, S., Lu, H.-C., Lin, C.-R., 1994. Tree size diversity and economic returns in uneven-aged forest stands. *For. Sci.* 40 (1), 83–103. <https://doi.org/10.1093/forestscience/40.1.83>.
- Burmeister, J.-M., Tockner, A., Reder, S., Engel, M., Richter, R., Mund, J.-P., Döllner, J., 2025. treeX: Unsupervised Tree Instance Segmentation in Dense Forest Point Clouds. *arXiv*. <https://arxiv.org/abs/2509.03633>.
- Cabo, C., Ordoñez, C., García-Cortés, S., Martínez, J., 2014. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* 87, 47–56. <https://doi.org/10.1016/j.isprsjprs.2013.10.008>.
- Cabo, C., Ordoñez, C., Sánchez-Lasheras, F., Roca-Pardiñas, J., de Cos-Juez, J., 2019. Multiscale supervised classification of point clouds with urban and forest applications. *Sensors* 19 (20), 4523. <https://doi.org/10.3390/s19204523>.
- Calders, K., Newnham, G., Burt, A., Murphy, S., Raunonen, P., Herold, M., Culvenor, D., Avitabile, V., Disney, M., Armston, J., Kaasalainen, M., 2015. Nondestructive estimates of above-ground biomass using terrestrial laser scanning. *Methods Ecol. Evol.* 6 (2), 198–208. <https://doi.org/10.1111/2041-210X.12301>.
- Calders, K., Adams, J., Armston, J., Bartholomeus, H., Bauwens, S., Bentley, L.P., Chave, J., Danson, F.M., Demol, M., Disney, M., Gaulton, R., Krishna Moorthy, S.M., Levick, S.R., Saarinen, N., Schaaf, C., Stovall, A., Terry, L., Wilkes, P., Verbeeck, H., 2020. Terrestrial laser scanning in forest ecology: Expanding the horizon. *Remote Sens. Environ.* 251, 112102. <https://doi.org/10.1016/j.rse.2020.112102>.
- Camuffo, E., Mari, D., Milani, S., 2022. Recent advancements in learning algorithms for point clouds: an updated overview. *Sensors* 22 (4), 1357. <https://doi.org/10.3390/s22041357>.
- Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J., 2023. LargeKernel3D: Scaling up kernels in 3D sparse CNNs. *arXiv*. <https://arxiv.org/abs/2206.10555>.
- Chen, Z., Xu, H., Chen, W., Zhou, Z., Xiao, H., Sun, B., Xie, X., & Kang, W. (2024). PointDC: Unsupervised semantic segmentation of 3D point clouds via cross-modal distillation and super-voxel clustering. *arXiv*. <https://arxiv.org/abs/2304.08965>.
- Cherlet, W., Dayal, K., Chen, S., Cooper, Z., Disney, M., Hanzl, A., Levick, S., Nightingale, J., Origo, N., Senf, C., Soenens, L., Terry, L., Van den Broeck, W.A.J., Calders, K., 2026. Benchmarking tree instance segmentation of terrestrial laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* 231, 230–247. <https://doi.org/10.1016/j.isprsjprs.2025.10.033>.
- Chu, X., Tian, Z., Zhang, B., Wang, X., & Shen, C. (2021). Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.
- Dai, W., Jiang, Y., Zeng, W., Chen, R., Xu, Y., Zhu, N., Xiao, W., Dong, Z., Guan, Q., 2023. MDC-Net: a multi-directional constrained and prior assisted neural network for wood and leaf separation from terrestrial laser scanning. *Int. J. Digital Earth* 16 (1), 1224–1245. <https://doi.org/10.1080/17538947.2023.2198261>.
- Disney, M., 2019. Terrestrial LiDAR: a three-dimensional revolution in how we look at trees. *New Phytol.* 222 (4), 1736–1741. <https://doi.org/10.1111/nph.15517>.
- García, M., Gajardo, J., Riano, D., Zhao, K., Martín, P., Ustin, S., 2015. Canopy clumping appraisal using terrestrial and airborne laser scanning. *Remote Sens. Environ.* 161, 78–88. <https://doi.org/10.1016/j.rse.2015.01.030>.
- Gonzalez de Tanago, J., Lau, A., Bartholomeus, H., Herold, M., Avitabile, V., Raunonen, P., Martius, C., Goodman, R.C., Disney, M., Manuri, S., Burt, A., Calders, K., 2018. Estimation of above-ground biomass of large tropical trees with terrestrial LiDAR. *Methods Ecol. Evol.* 9 (2), 223–234. <https://doi.org/10.1111/2041-210X.12904>.
- Hackel, T., Wegner, J.D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* III-3, 177–184. <https://doi.org/10.5194/isprs-annals-III-3-177-2016>.
- Han, T., Sánchez-Azofeifa, G.A., 2022. A Deep Learning Time Series Approach for Leaf and Wood Classification from Terrestrial LiDAR Point Clouds. *Remote Sens. (Basel)* 14 (13). <https://doi.org/10.3390/rs14133157>.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- 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. *Eco. Inform.* 84, 102888. <https://doi.org/10.1016/j.ecoinf.2024.102888>.
- Hilbert, D., 1935. Über die stetige Abbildung einer Linie auf ein Flächenstück. In: *Dritter Band: Analysis, Grundlagen Der Mathematik, Physik, Verschiedenes: Nebst Einer Lebensgeschichte*. Julius Springer, pp. 1–2.
- 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. *arXiv*. <https://arxiv.org/abs/1911.11236>.

- Hui, Z., Jin, S., Xia, Y., Wang, L., Yevenyo Ziggah, Y., Cheng, P., 2021. Wood and leaf separation from terrestrial LiDAR point clouds based on mode points evolution. *ISPRS J. Photogramm. Remote Sens.* 178, 219–239. <https://doi.org/10.1016/j.isprsjprs.2021.06.012>.
- Janvier, R., Cabo, C., 2024. Dendroptimized: Optimized C++ algorithms for dendromatics. *GitHub*. <https://github.com/3DFin/dendroptimized>.
- Jiang, T., Zhang, Q., Liu, S., Liang, C., Dai, L., Zhang, Z., Sun, J., Wang, Y., 2023. LWSNet: a Point-based segmentation network for leaf-wood separation of individual trees. *Forests* 14 (7), 1303. <https://doi.org/10.3390/f14071303>.
- Kaijalainen, R., Kukko, A., El Issaoui, A., Hyypä, J., Kaartinen, H., 2022. Semantic segmentation of point cloud data using raw laser scanner measurements and deep neural networks. *ISPRS Open Journal of Photogrammetry and Remote Sensing* 3, 100011. <https://doi.org/10.1016/j.ophoto.2021.100011>.
- Kankare, V., Holopainen, M., Vastaranta, M., Puttonen, E., Yu, X., Hyypä, J., Vaaja, M., Hyypä, H., Alho, P., 2013. Individual tree biomass estimation using terrestrial laser scanning. *ISPRS J. Photogramm. Remote Sens.* 75, 64–75. <https://doi.org/10.1016/j.isprsjprs.2012.10.003>.
- Kankare, V., Vauhkonen, J., Tanhuanpää, T., Holopainen, M., Vastaranta, M., Joensuu, M., Krooks, A., Hyypä, J., Hyypä, H., Alho, P., Viitala, R., 2014. Accuracy in estimation of timber assortments and stem distribution - a comparison of airborne and terrestrial laser scanning techniques. *ISPRS J. Photogramm. Remote Sens.* 97, 89–97. <https://doi.org/10.1016/j.isprsjprs.2014.08.008>.
- Kim, D.H., Ko, C.U., Kim, D.G., Kang, J.T., Park, J.M., Cho, H.J., 2023. Automated segmentation of individual tree structures using deep learning over LiDAR point cloud data. *Forests* 14 (6), 1159. <https://doi.org/10.3390/f14061159>.
- Krisanski, S., Taskhiri, M.S., Aracil, S.G., Herries, D., Muneri, A., Gurung, M.B., Montgomery, J., Turner, P., 2021. Forest structural complexity tool—an open source, fully-automated tool for measuring forest point clouds. *Remote Sens.* (Basel) 13 (22). <https://doi.org/10.3390/rs13224677>.
- Kulicki, M., Cabo, C., Trzcinski, T., Będkowski, J., Stereńczak, K., 2024. Artificial intelligence and terrestrial point clouds for forest monitoring. *Curr. For. Rep.* 11 (1), 5. <https://doi.org/10.1007/s40725-024-00234-4>.
- Laino, D., Cabo, C., Prendes, C., Janvier, R., Ordóñez, C., Nikonovas, T., Doerr, S., Santin, C., 2024. 3DFin: a software for automated 3D forest inventories from terrestrial point clouds. *Forestry: an International Journal of Forest Research* 97 (4), 479–496. <https://doi.org/10.1093/forestry/cpae020>.
- Laino, D., Cabo, C., Ordóñez, C., Bolanos, R., Janvier, R., Giulioni, F., Herrmann, M., Hudak, A., Parsons, R., Santin, C., 2025. SegmentedForests: a labelled dataset of terrestrial LiDAR point clouds for semantic segmentation of forests. *Forestry: an International Journal of Forest Research* cpaf062. <https://doi.org/10.1093/forestry/cpaf062>.
- Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4558–4567. <https://doi.org/10.1109/CVPR.2018.00479>.
- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521 (7553), 436–444. <https://doi.org/10.1038/nature14539>.
- Lévy, B., Fillois, A., 2015. Geogram: a library for geometric algorithms. In *ADAMOS 2015*, 45.
- Li, Y., Guo, Q., Su, Y., Tao, S., Zhao, K., Xu, G., 2017. Retrieving the gap fraction, element clumping index, and leaf area index of individual trees using single-scan data from a terrestrial laser scanner. *ISPRS J. Photogramm. Remote Sens.* 130, 308–316. <https://doi.org/10.1016/j.isprsjprs.2017.06.006>.
- Li, G., Müller, M., Ghanem, B., & Koltun, V. (2021). Training graph neural networks with 1000 layers. In *Proceedings of the International Conference on Machine Learning (ICML)* (Vol. 139, pp. 6437–6449). PMLR.
- Lines, E.R., Allen, M., Cabo, C., Calders, K., Debus, A., Grieve, S.W.D., Miltiadou, M., Noach, A., Owen, H.J.F., Puliti, S., 2022. AI Applications in Forest Monitoring Need Remote Sensing Benchmark Datasets. *arXiv*. <https://arxiv.org/abs/2212.09937>.
- Liu, B., Wang, M., Foroosh, H., Tappen, M., & Penky, M. (2015). Sparse convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 806–814). IEEE. <https://doi.org/10.1109/CVPR.2015.7298681>.
- Liu, J., Wang, T., Skidmore, A.K., Jones, S., Heurich, B., Beudert, B., Premier, J., 2019. Comparison of terrestrial LiDAR and digital hemispherical photography for estimating leaf angle distribution in European broadleaf beech forests. *ISPRS J. Photogramm. Remote Sens.* 158, 76–89. <https://doi.org/10.1016/j.isprsjprs.2019.09.015>.
- Liu, J., Yu, Z., Breckon, T.P., Shum, H.P.H., 2023. U3DS3: Unsupervised 3D Semantic Scene Segmentation. *arXiv*. <https://arxiv.org/abs/2311.06018>.
- Liu, J., Wang, D., Gong, H., Wang, C., Zhu, J., Wang, D., 2026. A synthetic data generation framework for deep learning-based LiDAR forest structure analysis. *Remote Sens. Environ.* 341, 115436. <https://doi.org/10.1016/j.rse.2026.115436>.
- Loudermilk, E.L., Hiers, J.K., O'Brien, J.J., Mitchell, R.J., Singhania, A., Fernandez, J.C., Cropper, W.P., Slatton, K.C., 2009. Ground-based LiDAR: a novel approach to quantify fine-scale fuelbed characteristics. *Int. J. Wildland Fire* 18 (6), 676–685. <https://doi.org/10.1071/WF07138>.
- Lu, H., Li, B., Yang, G., Fan, G., Wang, H., Pang, Y., Wang, Z., Lian, Y., Xu, H., Huang, H., 2025. Towards a point cloud understanding framework for forest scene semantic segmentation across forest types and sensor platforms. *Remote Sens. Environ.* 318, 114591. <https://doi.org/10.1016/j.rse.2024.114591>.
- Ma, L., Zheng, G., Eitel, J.U.H., Moskal, L.M., He, W., Huang, H., 2016. Improved Salient Feature-based Approach for Automatically Separating Photosynthetic and Nonphotosynthetic Components within Terrestrial Lidar Point Cloud Data of Forest Canopies. *IEEE Trans. Geosci. Remote Sens.* 54 (2), 679–696. <https://doi.org/10.1109/TGRS.2015.2459716>.
- Ma, Z., Dong, Y., Zi, J., Xu, F., Chen, F., 2023. Forest-PointNet: a deep learning model for vertical structure segmentation in complex forest scenes. *Remote Sens. (Basel)* 15 (19), 4793. <https://doi.org/10.3390/rs15194793>.
- McGee, G.G., Leopold, D.J., Nyland, R.D., 1999. Structural characteristics of old-growth, maturing, and partially cut northern hardwood forests. *Ecol. Appl.* 9 (4), 1316–1329. <https://doi.org/10.2307/2641399>.
- Morton, G.M., 1966. A computer oriented geodetic data base and a new technique in file sequencing. *IBM Ltd.*
- Olschofsky, K., Mues, V., Köhl, M., 2016. Operational assessment of aboveground tree volume and biomass by terrestrial laser scanning. *Comput. Electron. Agric.* 127, 699–707. <https://doi.org/10.1016/j.compag.2016.07.030>.
- Olsoy, P.J., Glenn, N.F., Clark, P.E., Derryberry, D.W.R., 2014. Aboveground total and green biomass of dryland shrub derived from terrestrial laser scanning. *ISPRS J. Photogramm. Remote Sens.* 88, 166–173. <https://doi.org/10.1016/j.isprsjprs.2013.12.006>.
- Oviedo de la Fuente, M., Cabo, C., Roca-Pardiñas, J., Loudermilk, E.L., Ordóñez, C., 2024. 3D point cloud semantic segmentation through functional data analysis. *J. Agric. Biol. Environ. Stat.* 29 (4), 723–744. <https://doi.org/10.1007/s13253-023-00567-w>.
- Pointcept Contributors (2023). Pointcept: A codebase for point cloud perception research [Computer software]. *GitHub*. <https://github.com/Pointcept/Pointcept>.
- Peng, B., Wu, X., Jiang, L., Chen, Y., Zhao, H., Tian, Z., & Jia, J. (2024). OA-CNNs: Omni-adaptive sparse CNNs for 3D semantic segmentation. *arXiv*. <https://arxiv.org/abs/2403.14418>.
- Puliti, S., Pearce, G., Surový, 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*. <https://arxiv.org/abs/2309.01279>.
- Putman, E.B., Popescu, S.C., Eriksson, M., Zhou, T., Klockow, P., Vogel, J., Moore, G.W., 2018. Detecting and quantifying standing dead tree structural loss with reconstructed tree models using voxelized terrestrial lidar data. *Remote Sens. Environ.* 209, 52–65. <https://doi.org/10.1016/j.rse.2018.02.028>.
- Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). PointNet: Deep learning on point sets for 3D classification and segmentation. *arXiv*. <https://arxiv.org/abs/1612.00593>.
- Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017b). PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv*. <https://arxiv.org/abs/1706.02413>.
- Qian, G., Hammoud, H., Li, G., Thabet, A., Ghanem, B., 2021. ASSANet: an anisotropic separable set abstraction for efficient point cloud representation learning. *Advances in Neural Information Processing Systems (NeurIPS)* 34.
- Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H.A.A.K., Elhoseiny, M., Ghanem, B., 2022. PointNetXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. *arXiv*. <https://arxiv.org/abs/2206.04670>.
- Robert, D., Raguet, H., Landrieu, L., 2023. Efficient 3D semantic segmentation with Superpoint Transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Rozenberszki, D., Litany, O., Dai, A., 2024. UnScene3D: Unsupervised 3D Instance Segmentation for Indoor Scenes. *arXiv*. <https://arxiv.org/abs/2303.14541>.
- Şahin, E., Arslan, N.N., Özdemir, D., 2025. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Comput. & Applic.* 37, 859–965. <https://doi.org/10.1007/s00521-024-10437-2>.
- Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., Sattarvand, J., 2024. A comprehensive overview of deep learning techniques for 3D point cloud classification and semantic segmentation. *Mach. Vis. Appl.* 35 (4), 67. <https://doi.org/10.1007/s00138-024-01543-1>.
- Schraik, D., Hovi, A., Rautiainen, M., 2021. Crown level clumping in Norway spruce from terrestrial laser scanning measurements. *Agric. For. Meteorol.* 296. <https://doi.org/10.1016/j.agrformet.2020.108238>.
- Shen, X., Huang, Q., Wang, X., Li, J., Xi, B., 2022. A deep learning-based method for extracting standing wood feature parameters from terrestrial laser scanning point clouds of artificially planted forest. *Remote Sens. (Basel)* 14 (15). <https://doi.org/10.3390/rs141513842>.
- SPConv Contributors, 2022. Spconv: Spatially Sparse Convolution Library [Computer software]. *GitHub*. <https://github.com/traveller59/spconv>.
- Stovall, A.E.L., MacFarlane, D.W., Crawford, D., Jovanovic, T., Frank, J., Brack, C., 2023. Comparing mobile and terrestrial laser scanning for measuring and modelling tree stem taper. *Forestry: an International Journal of Forest Research* 96 (5), 705–717. <https://doi.org/10.1093/forestry/cpad012>.
- Straker, A., Puliti, S., Breidenbach, J., Klein, C., Pearce, G., Astrup, R., Magdon, P., 2023. Instance segmentation of individual tree crowns with YOLOv5: a comparison of approaches using the ForInstance benchmark LiDAR dataset. *ISPRS Open Journal of Photogrammetry and Remote Sensing* 9, 100045. <https://doi.org/10.1016/j.ophoto.2023.100045>.
- Tao, S., Guo, Q., Xu, S., Su, Y., Li, Y., Wu, F., 2015. A geometric method for wood-leaf separation using terrestrial and simulated lidar data. *Photogramm. Eng. Remote Sens.* 81 (10), 767–776. <https://doi.org/10.14358/PERS.81.10.767>.
- Tian, J., Li, H., Sun, X., Zhou, Y., Ma, W., Chen, J., Zhang, J., Xu, Y., 2023. Quality assessment of shrub observation data based on TLS: a case of revegetated shrubland, Southern Qinghai-Tibetan Plateau. *Land Degrad. Dev.* 34 (5), 1570–1581. <https://doi.org/10.1002/ldr.4554>.
- 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 (NeurIPS)*, 30. <https://arxiv.org/abs/1706.03762>.
- Vicari, M.B., Disney, M., Wilkes, P., Burt, A., Calders, K., Woodgate, W., 2019. Leaf and wood classification framework for terrestrial LiDAR point clouds. *Methods Ecol. Evol.* 10 (5), 680–694. <https://doi.org/10.1111/2041-210X.13144>.

- Wang, D., Brunner, J., Ma, Z., Lu, H., Hollaus, M., Pang, Y., Pfeifer, N., 2018. Separating tree photosynthetic and non-photosynthetic components from point cloud data using Dynamic Segment Merging. *Forests* 9 (5). <https://doi.org/10.3390/f9050252>.
- Wang, D., Momo Takoudjou, S., Casella, E., 2019. LeWoS: a universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR. *Methods Ecol. Evol.* 11 (3), 376–389. <https://doi.org/10.1111/2041-210X.13342>.
- Wang, D., 2020. Unsupervised semantic and instance segmentation of forest point clouds. *ISPRS J. Photogramm. Remote Sens.* 165, 86–97. <https://doi.org/10.1016/j.isprsjprs.2020.04.020>.
- 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 Sens. Environ.* 313, 114367. <https://doi.org/10.1016/j.rse.2024.114367>.
- Wikum, D.A., Shanholtzer, G.F., 1978. Application of the Braun-Blanquet cover-abundance scale for vegetation analysis in land development studies. *Environ. Manag.* 2 (4), 323–329. <https://doi.org/10.1007/bf01866672>.
- Winiwarter, L., Esmoris Pena, A.M., Weiser, H., Anders, K., Martínez Sánchez, J., Searle, M., Höfle, B., 2022. Virtual laser scanning with HELIOS++: a novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. *Remote Sens. Environ.* 269, 112772. <https://doi.org/10.1016/j.rse.2021.112772>.
- Wu, B., Zheng, G., Chen, Y., 2020. An improved convolution neural network-based model for classifying foliage and woody components from terrestrial laser scanning data. *Remote Sens. (Basel)* 12 (6), 1010. <https://doi.org/10.3390/rs12061010>.
- 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 (NeurIPS)*.
- 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. In: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851.
- Xi, Z., Chasmer, L., Hopkinson, C., 2023. Delineating and reconstructing 3D forest fuel components and volumes with terrestrial laser scanning. *Remote Sens. (Basel)* 15 (19), 4778. <https://doi.org/10.3390/rs15194778>.
- Xi, Z., Degenhardt, D., 2025. A new unified framework for supervised 3D crown segmentation (TreeisoNet) using deep neural networks across airborne, UAV-borne, and terrestrial laser scans. *ISPRS Open Journal of Photogrammetry and Remote Sensing* 15, 100083. <https://doi.org/10.1016/j.ophoto.2025.100083>.
- Xiang, B., Wielgosz, M., Kontogianni, T., Peters, T., Puliti, S., Astrup, R., Schindler, K., 2024. Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. *Remote Sens. Environ.* 305, 114078. <https://doi.org/10.1016/j.rse.2024.114078>.
- Xiang, B., Wielgosz, M., Puliti, S., Král, K., Krůček, M., Missarov, A., & Astrup, R., 2025. ForestFormer3D: A unified framework for end-to-end segmentation of forest LiDAR 3D point clouds. *arXiv*. [arXiv:2506.16991](https://arxiv.org/abs/2506.16991).
- Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L., 2022. Scaling vision transformers. In: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12104–12113.
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens. (Basel)* 8 (6), 501. <https://doi.org/10.3390/rs8060501>.
- Zhang, Z., Yang, B., Wang, B., & Li, B. (2023). GrowSP: Unsupervised semantic segmentation of 3D point clouds. *arXiv*. <https://arxiv.org/abs/2305.16404>.
- Zhang, Z., Ding, J., Jiang, L., Dai, D., & Xia, G.-S. (2024). FreePoint: Unsupervised point cloud instance segmentation. *arXiv*. <https://arxiv.org/abs/2305.06973>.
- Zhao, H., Jiang, L., Jia, J., Torr, P.H.S., Koltun, V., 2021. Point Transformer. In: *In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16259–16268. <https://doi.org/10.1109/ICCV48922.2021.01598>.