forests

MDPI

Article

TreeDBH: Dual Enhancement Strategies for Tree Point Cloud Completion in Medium–Low Density UAV Data

Yunlian Su [1], Zhibo Chen [1,2,*] and Xiaojing Xue [1]

[1] School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China; syl3220299@bjfu.edu.cn (Y.S.); xuexiaojing@bjfu.edu.cn (X.X.)
[2] Engineering Research Center for Forestry-Oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China
* Correspondence: zhibo@bjfu.edu.cn

Abstract: Medium–low density UAV point clouds often suffer from incomplete lower canopy structures and sparse distributions due to self-occlusion. While existing point cloud completion models achieve high metric accuracy, they inadequately address missing regions in trunks and lower canopy areas. To resolve these issues, this paper proposes a hierarchical random sampling strategy and a spatially constrained loss function. First, we dynamically stratify point clouds based on density distribution characteristics, employing hierarchical random sampling to preserve proportional representation of lower-level points, thereby effectively retaining basal tree structure information. Second, we introduce a distance constraint term for mid-lower point clouds into the symmetrical Chamfer distance (CD) loss, compelling models to prioritize completion quality in trunk base regions. Experiments on the FOR-instance-created completion dataset and Xiong'an dataset demonstrate that our method significantly enhances structural recovery capability at tree trunk bases, with visual results outperforming the baseline SeedFormer model. Additionally, we refer to existing point cloud-based diameter at breast height (DBH) calculation methods to measure the completed trees and compare the computed results with the measured values to evaluate the accuracy of the completion effect. Experimental results show that, after integrating our proposed strategies with existing completion methods, the accuracy of DBH measurement from point clouds is significantly improved. This study provides novel insights for addressing structural bias in tree point cloud completion and offers valuable references for digital forestry resource management.

Keywords: UAV point clouds; point cloud completion; deep learning

check for updates

Academic Editor: Nikolay S. Strigul

Received: 4 March 2025
Revised: 27 March 2025
Accepted: 9 April 2025
Published: 11 April 2025

Citation: Su, Y.; Chen, Z.; Xue, X. TreeDBH: Dual Enhancement Strategies for Tree Point Cloud Completion in Medium–Low Density UAV Data. *Forests* **2025**, *16*, 667. https://doi.org/10.3390/f16040667

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Point cloud technology, as a three-dimensional coordinate-based digitalization method, enables rapid acquisition of tree morphological features, structural characteristics, and spatial distribution patterns through sensor-captured arboreal point cloud data [1,2]. Unmanned Aerial Vehicles (UAVs) equipped with LiDAR systems demonstrate exceptional capability in large-scale forest point cloud data collection [3–5]. Compared to conventional ground surveys, UAV platforms can rapidly survey extensive areas while overcoming accessibility limitations in remote regions and complex terrains. However, the inherent geometric complexity of trees and environmental constraints during scanning, including equipment precision limitations, flight altitude restrictions, self-occlusion within tree canopies, and occlusion from surrounding vegetation, frequently result in data incompleteness, particularly in lower canopy regions, inter-tree spaces, and dense forest stands,

point clouds often exhibit sparsity issues. To address data incompleteness and sparsity in medium–low density UAV point clouds, deep learning-based point cloud completion methods [6–8] offer promising solutions for reconstructing missing segments, thereby delivering more comprehensive and precise 3D tree representations. Enhanced point cloud integrity significantly improves the accuracy of extracting critical biophysical parameters [9–11] including tree height, diameter at breast height (DBH), and crown projection area. These parameters hold substantial application value for forest resource monitoring, ecological environment studies, and sustainable forestry management.

Point cloud completion methodologies can be categorized into traditional approaches and deep learning-based techniques [12]. Traditional methods [13–17] infer missing regions by leveraging the local geometric features of point clouds. Common geometric reconstruction strategies include neighborhood-based interpolation, Poisson reconstruction, and triangular mesh generation. These approaches necessitate inherent symmetry and regularity in raw point clouds, often yielding significant errors when handling large missing areas or complex point distributions. In recent years, deep learning-based point cloud completion has emerged as a prominent research focus. Predominant frameworks encompass voxel-based [18–20], view-based [21–24], and point-based [25–27] paradigms. Voxel-based methods reconstruct missing regions by analyzing structural relationships among volumetric pixels, while view-based methods infer complete models through multi-view projections generated from varying perspectives. The advent of PointNet [28] enabled direct processing of raw point clouds via deep neural networks, marking a paradigm shift. The PCN [29] network pioneered point-based completion frameworks by integrating the architectural principles of FoldingNet [30] and PointNet, employing a coarse-to-fine decoder to predict complete point clouds. Subsequent advancements include SANet [31], which enhanced completion performance through skip-attention mechanisms, and PF-Net [32], which improved precision via multi-resolution feature extraction, hierarchical generation, and adversarial discriminators for quality assessment. Transformer architectures [33] demonstrate inherent suitability for processing unordered point cloud data through permutation-invariant attention mechanisms, effectively modeling global point relationships. The Pointr [34] framework represents incomplete point clouds as point proxies, utilizing transformer-based [35] encoders–decoders for structural restoration. Recent innovations like SnowflakeNet [36] implement deconvolutional upsampling with skip-Transformer modules across decoding layers to progressively refine missing regions while preserving local geometric features.

While existing point cloud completion methods have achieved notable progress in global structure modeling and local detail recovery through ongoing research, most approaches predominantly rely on synthetic datasets characterized by planar surfaces and uniform point density. Furthermore, the simulation of missing regions typically employs random spherical elimination [32,34], where points within a randomly positioned sphere are removed. However, in real-world scenarios, missing regions in UAV-captured tree point clouds predominantly result from occlusion by foliage and branches. Limited by LiDAR penetration capabilities, point density gradually diminishes near ground surfaces, exacerbating density disparities between upper and lower canopy layers when applying current completion methods to practical cases, ultimately compromising completion quality. To address this challenge, this study proposes a hierarchical random sampling strategy tailored to authentic UAV point cloud characteristics, with SeedFormer [37] as the baseline framework. The strategy initiates with statistical analysis of lower-trunk point density, calculating its proportional representation within the complete point cloud and determining the median proportion threshold. During sampling, it prioritizes retaining lower-level points above this median threshold, ensuring moderate input density for trunk

base regions at the data level and mitigating adverse effects from excessive sparsity on completion performance. Additionally, we enhance model attention to lower canopy structures by incorporating the Chamfer distance [38] between completed point clouds and ground-truth lower-trunk regions into the loss function. Through these dual strategies, the model achieves effective trunk structure completion even under low-density lower canopy conditions, significantly improving point cloud integrity. This advancement enables more structurally coherent and ecologically plausible tree reconstructions.

Terrestrial laser scanning (TLS)-acquired point clouds exhibit high quality with distinct trunk profiles, enabling accurate DBH estimation. In contrast, most UAV-derived tree point clouds show degraded quality at DBH measurement heights (1.3–1.5 m), which is often insufficient to support reliable DBH computation. In traditional forestry research, DBH estimation methods based on allometric equations [39] are widely used. These methods rely on tree growth models and empirical parameters, providing relatively stable DBH estimates to a certain extent. To reduce the workload of manual measurements, we aim to improve the automation level of DBH measurement using UAV point cloud data. Jie Xu et al. [40] employed the BlendMask algorithm to accurately delineate tree crown contours in UAV imagery and introduced a Bayesian neural network to establish a correlation model between individual tree crown size and DBH. The CaR3DMIC method [41] directly extracts key tree features, such as crown height and crown area, from image data. However, in environments with medium- and low-density point cloud data, the complexity of tree morphology and occlusion effects make it challenging to extract other critical parameters directly, leading to difficulties in ensuring the accuracy of DBH measurement. Compared to traditional allometric equations, point cloud-based methods can reconstruct tree trunk structures in a data-driven manner, reducing reliance on empirical formulas and minimizing systematic errors. Furthermore, our approach focuses on improving the overall quality of tree structure completion, particularly the integrity of lower regions. DBH measurement based on point clouds provides a more intuitive reflection of the effectiveness of our completion method. Therefore, we measure DBH using point clouds [42] and introduce quality evaluation metrics to qualitatively analyze the accuracy of trunk completion. We conducted single-tree point cloud completion experiments on both our proprietary dataset and the adjusted FOR-instance dataset [43]. Experimental results demonstrate that the adjusted model maintains completion performance while more accurately restoring tree contours, significantly enhancing DBH measurement accuracy.

Our primary contributions are threefold:

(1) This study established a proprietary dataset through UAV-mounted LiDAR and terrestrial laser scanning (TLS) data acquisition, while constructing a dedicated point cloud completion benchmark based on the FOR-instance dataset;
(2) To address self-occlusion-induced incompleteness and canopy layer density disparities in medium–low density UAV point clouds, we proposed two enhancement strategies that significantly improve the integrity and precision of tree point cloud completion;
(3) In this study, we applied a DBH measurement method to fit the DBH of the completed point clouds and compare the results with ground-truth measurements, verifying the impact and value of point cloud completion on single-tree parameter extraction.

2. Materials

2.1. Point Cloud Completion Dataset Construction Using the FOR-Instance Benchmark

2.1.1. Overview of the FOR-Instance Dataset

The FOR-instance dataset [43] (available at https://doi.org/10.5281/zenodo.8287792 accessed on 27 August 2024) comprises UAV LiDAR-derived point clouds collected from 30 forest plots across five countries. It features manually annotated semantic class labels

and tree IDs for individual points, with seven semantic categories: unclassified points, low vegetation, ground surface, incomplete tree points, trunks, live branches, and dead branches. Field-measured DBH values are also included for selected trees.

The dataset encompasses five forest types:

- NIBIO: Boreal conifer-dominated forests (Norway, 42% of annotated data) [44];
- CULS: Temperate conifer-dominated forests (Czech Republic, 5%) [45];
- TUWIEN: Deciduous floodplain forests (Austria, 29%) [46];
- RMIT: White peppermint eucalyptus-dominated stands (Australia, 11%);
- SCION: Monocultural radiata pine plantations (New Zealand, 13%).

FOR-instance captures realistic structural complexities, including curved trunks, occluded or sparsely sampled trunks, fallen trees, and understory vegetation. Notably, 89% of trees are visible from the upper canopy layer, while 11% reside in the sub-canopy stratum with obscured crowns. Geographical disparities manifest through distinct species compositions and terrain variations across regions. Significant interspecific differences in crown architecture are observed. Point cloud density varies substantially, ranging from 9629 pts/m^2 in NIBIO regions to 498 pts/m^2 in RMIT areas, collectively positioning FOR-instance as a high-density point cloud benchmark.

2.1.2. FOR-Instance Dataset Processing

We initially extracted all individual trees from the FOR-instance dataset using tree ID annotations, retaining specimens with over 16,384 points per tree as labeled point clouds to ensure high-density characteristics and model input compatibility. Building upon this foundation, we synthesized medium–low density UAV-like point clouds as model inputs by simulating real-world sensor limitations. Specifically, we simulated the self-occlusion phenomenon of LiDAR sensors and the uneven point cloud density between the upper and lower layers of the tree canopy by applying a weighted random downsampling method to process the point cloud data. Specifically, we first generated a viewpoint based on the center, diameter, and maximum height of the point cloud's bounding box. Then, using this viewpoint and a defined radius, we classified the points into visible and occluded categories. During the random downsampling process, we stochastically removed points based on deletion weights. Visible points were assigned lower deletion weights to preserve as much information as possible, while occluded points were given higher deletion weights to increase their probability of removal, thereby enhancing the missing rate of point clouds in occluded areas. This process constrained final point densities to 150–360 pts/m^2 across all plots, representative of practical UAV LiDAR survey conditions. Ultimately, input point clouds were rigorously paired with their corresponding labeled counterparts to ensure spatial correspondence. Figure 1 illustrates the visual comparison of downsampling effects across five representative regions.

Following rigorous screening, we obtained 607 qualified individual trees, with 80% (486 trees) allocated to the training set and 20% (121 trees) to the test set. The final dataset composition per region is detailed in Table 1. Due to the low initial point cloud acquisition density in the RMIT region, this area yielded the minimal dataset contribution, containing only four trees exceeding 16,384 points. Three RMIT samples were assigned to the training set and one to the test set through stratified random sampling across five geographic regions. To enhance model generalizability, we implemented data augmentation on the training set via geometric transformations including translation and random rotation about the centroid axis. Post-augmentation, the training set expanded to 3888 input samples while maintaining the test set size. The spatial distribution is visualized in Figure 2.

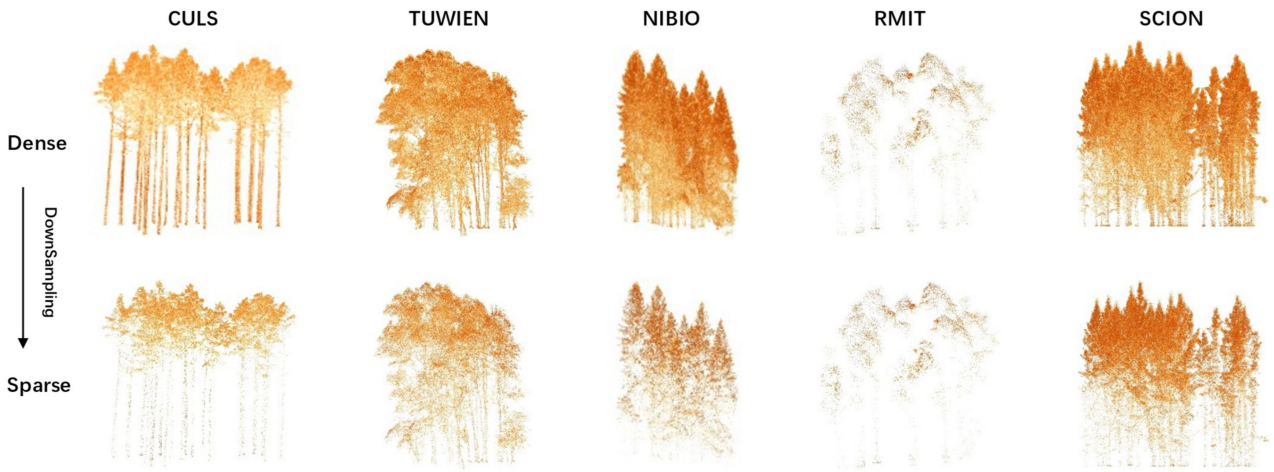


Figure 1. Downsampling effects visualization on the FOR-instance dataset.

Table 1. Training–testing split statistics of the FOR-instance point cloud completion dataset. In the "Training Set" column, the number before the multiplication sign represents the original number of trees before data augmentation. We applied an 8× data augmentation to the trees in the training set, resulting in a final training sample size that is eight times the original number. The number of trees in the test set remains unchanged and has not undergone data augmentation.

Region	TrainSet	TestSet
CULS	$37 \times 8 = 296$	10
NIBIO	$315 \times 8 = 2520$	81
RMIT	$3 \times 8 = 24$	1
SCION	$78 \times 8 = 624$	18
TEWIEN	$53 \times 8 = 424$	11
Total	$486 \times 8 = 3888$	121

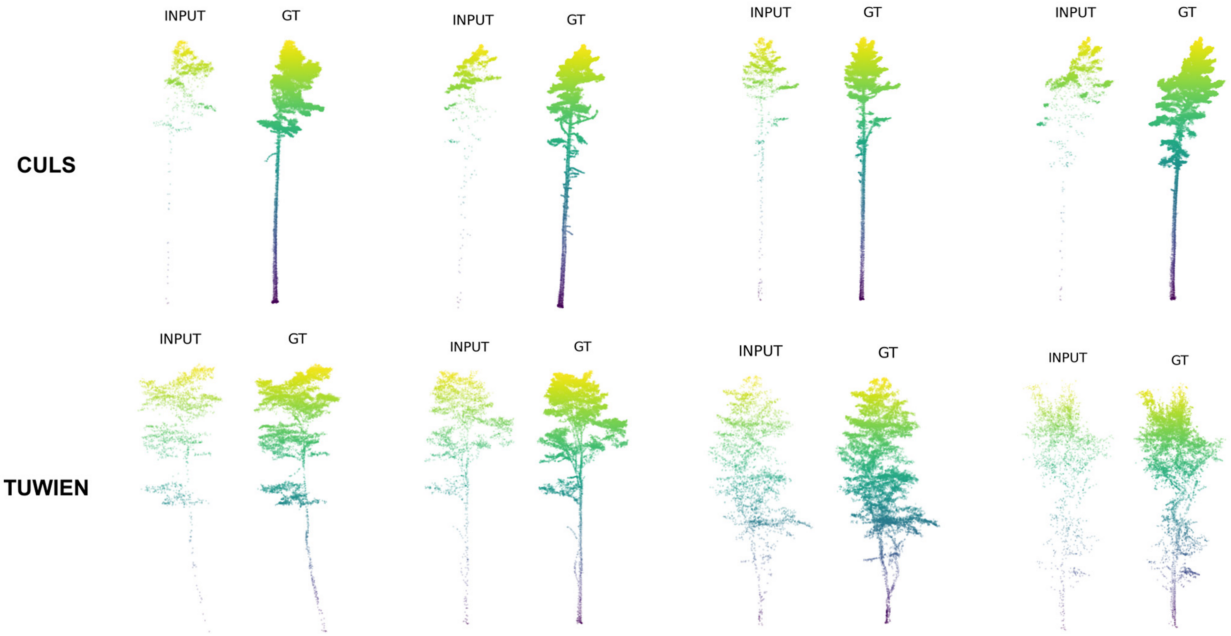


Figure 2. *Cont.*

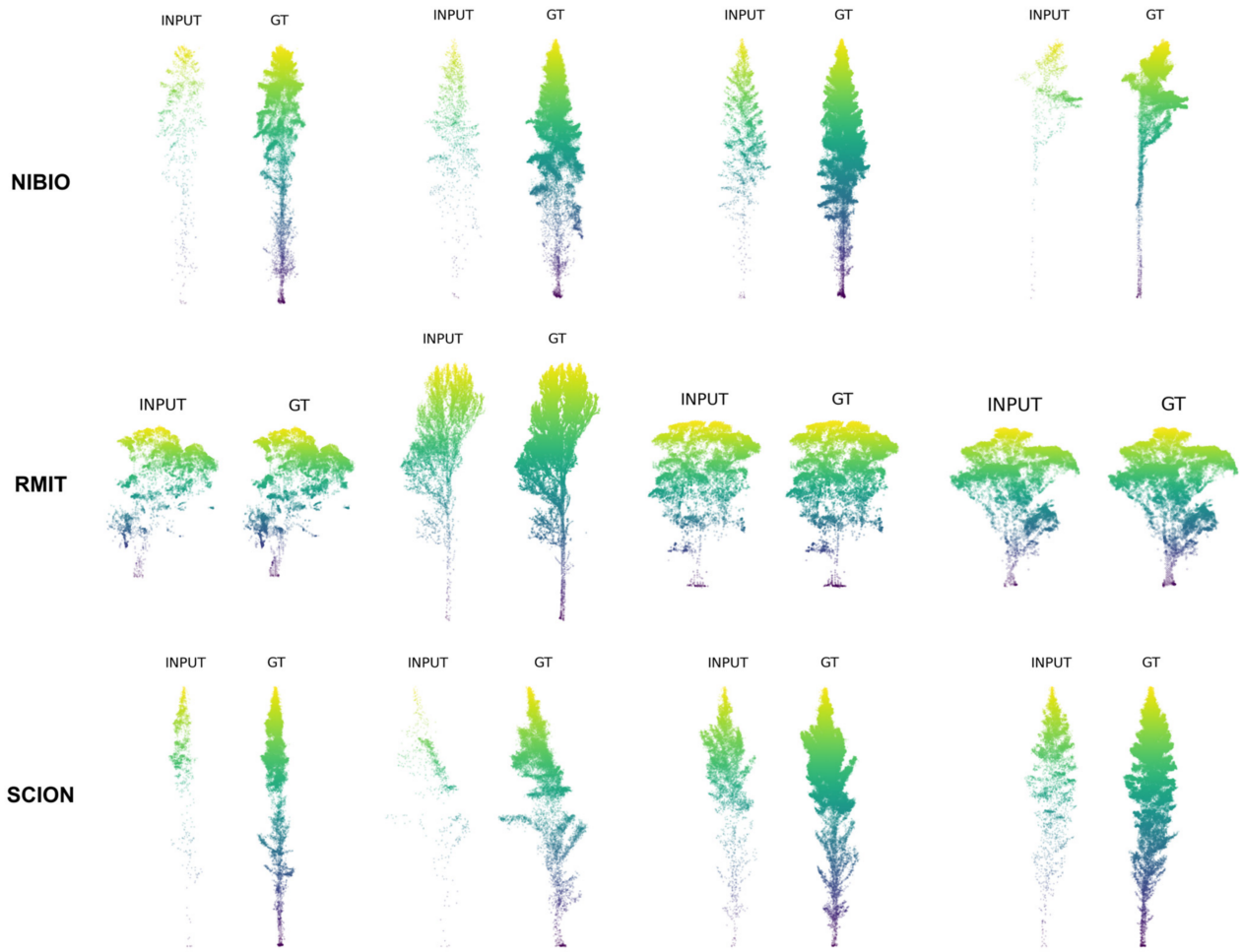


Figure 2. Representative samples from the FOR-instance completion dataset. "INPUT" represents the input point cloud from the dataset, while "GT" (ground truth) denotes the corresponding real tree point cloud. It is primarily used for model training, performance evaluation, and result validation. The image assigns colors to the trees based on their height.

2.2. Xiong'an Dataset

2.2.1. Study Area

The study area of this dataset is located in central Hebei Province, China (38°43′–39°10′ N, 115°38′–116°20′ E), encompassing Xiong County, Rongcheng County, Anxin County, and surrounding regions, as shown in Figure 3. The Xiong'an New Area lies in the mid-latitude zone and features a warm temperate monsoon continental climate with four distinct seasons. The annual average temperature is 11.7 °C, with the highest monthly average temperature of 26 °C in July and the lowest monthly average temperature of −4.9 °C in January. The region receives an average of 2685 h of sunshine per year and an annual precipitation of 551.5 mm, 80% of which falls between June and September. The frost-free period lasts approximately 185 days. The dominant tree species in this area include 13 major species, such as *Robinia pseudoacacia*, *Fraxinus chinensis*, *Ulmus pumila*, *Sophora japonica*, *Acer truncatum*, *Catalpa bungei*, *Ailanthus altissima*, *Juniperus chinensis*, *Pinus armandii*, *Prunus davidiana*, *Populus* spp., *Ginkgo biloba*, *Pinus tabuliformis*, *Platycladus orientalis*, *Platanus acerifolia*, *Catalpa ovata*, *Malus micromalus*, *Salix matsudana*, *Koelreuteria paniculata*, *Toona sinensis*, *Acer pictum*, among others. In this study, 13 experimental plots were selected within the Xiong'an region, covering eight dominant tree species: *Ginkgo biloba*, *Sophora japonica*, *Eucommia ulmoides*, *Fraxinus chinensis*, *Koelreuteria paniculata*, *Catalpa ovata*, *Ulmus* spp., and *Acer pictum*.

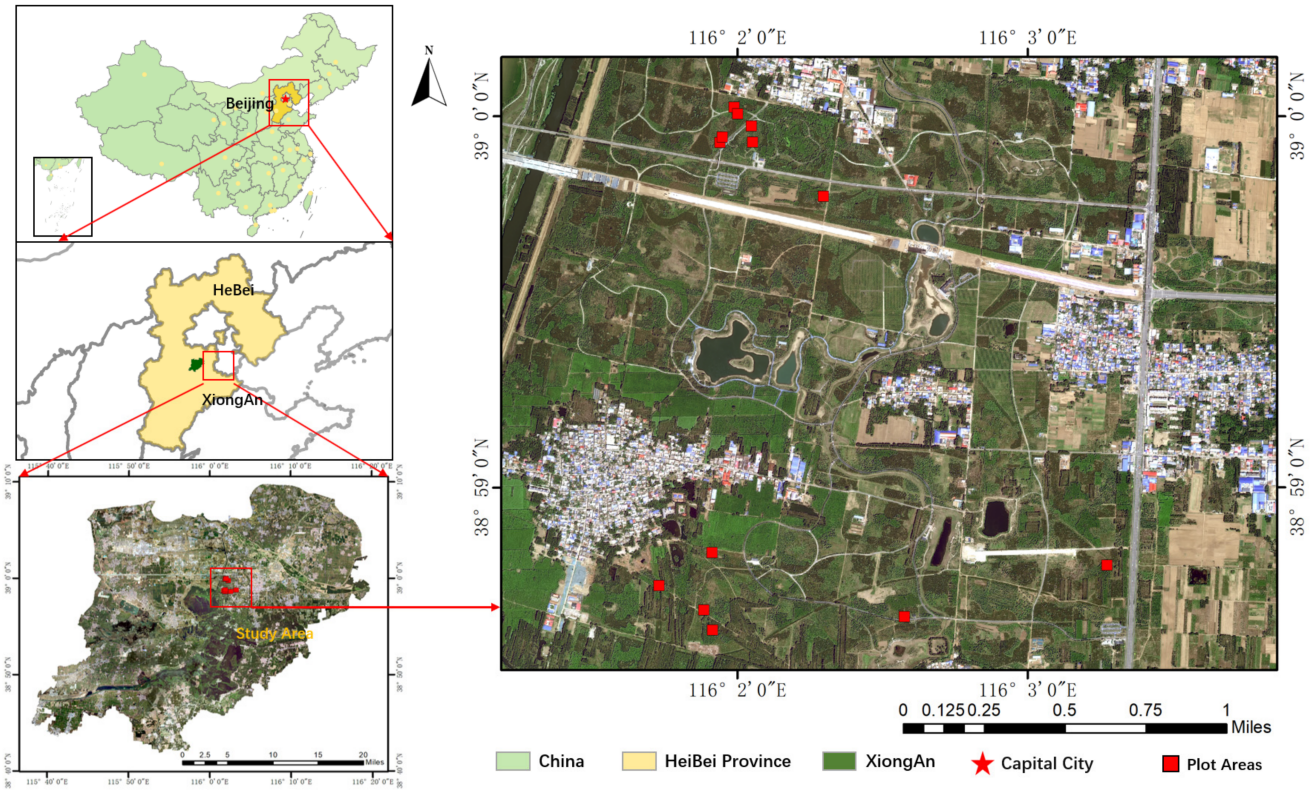


Figure 3. General overview of the study area. The study area is located within Xiong'an New Area, Hebei Province, China. The red squares indicate the locations of the experimental plots.

2.2.2. ULS Data

This study utilized the DJI Zenmuse L2 LiDAR module mounted on the DJI Matrice 350 UAV platform to acquire unmanned airborne laser scanning (ULS) data. The LiDAR system operates at a wavelength of 905 nm, with a ranging accuracy of 2 cm at a distance of 150 m. It features a laser divergence angle of 0.6 mrad (horizontal) × 0.2 mrad (vertical) and a pulse repetition frequency of 240 kHz. In multiple return mode, the system can acquire data at a maximum rate of 1.2 million points per second. Data collection was conducted from August to September 2024, during the morning hours under optimal lighting conditions. The flight speed was set to 18 m/s, and the flight altitude was maintained at 80 m above the takeoff point. The surveyed area covered 13 experimental plots included in the dataset, with point cloud densities ranging from 148.16 to 462.87 points per square meter (pts/m^2), and an average density of 260.28 pts/m^2 across all plots. After data acquisition, noise in the raw scan data was removed using Gaussian filtering. The point cloud was then classified into ground and non-ground points using the cloth simulation filtering algorithm. Finally, the point cloud data were normalized based on ground elevation.

2.2.3. TLS Data

We utilized the RIEGL VZ-600i laser scanner (RIEGL Laser Measurement Systems GmbH, Horn, Austria) for terrestrial laser scanning (TLS) data acquisition. This instrument features a scanning field of view of ≥360° × 105° (horizontal × vertical) and achieves a scanning accuracy of ≥5 mm at a distance of 100 m, with a laser pulse repetition rate of 2.2 million points per second. The TLS data collection was conducted concurrently with UAV-based data acquisition, covering 13 experimental plots, each measuring 20 m × 20 m. For each plot, three independent scans were performed. The acquired scans were subsequently processed using the accompanying software for automatic registration, noise removal, and point cloud normalization. To ensure data quality while improving process-

ing efficiency, we employed CloudCompare software (version 2.13.0) [47] and applied the octree-based sampling method to downsample the point cloud. After downsampling, the density of the ground point cloud ranged from 10,687.20 to 52,323.46 points per square meter (pts/m^2), with an average density of 36,771.18 pts/m^2 across the 13 plots. The acquired point cloud data are illustrated in the corresponding Figure 4.

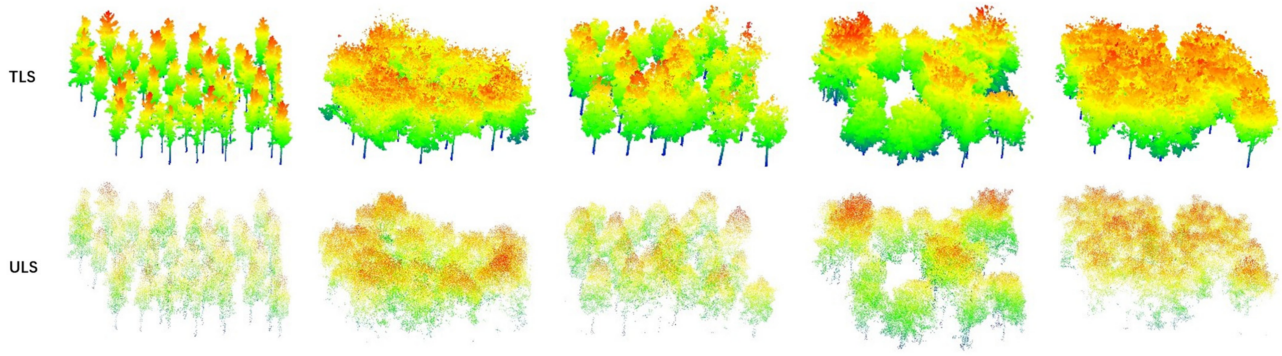


Figure 4. Partial visualization of sample plots in the Xiong'an collected dataset. The image assigns colors to the trees based on their height.

2.2.4. Field Data

The sample plot data collection was conducted concurrently with UAV-based LiDAR and terrestrial LiDAR data acquisition. The selection of sample plots considered multiple factors, including dominant local tree species, growth season, accessibility, and terrain, to ensure representativeness. An RTK device was used to ensure precise plot positioning and alignment, while standard measuring tapes were employed to measure distances within the plots. Each sample plot was set to a size of 20 m × 20 m, with its center coordinates recorded. Within each plot, we measured the DBH and geographic coordinates of individual trees. Following standard field measurement protocols, trees with a DBH greater than 5 cm were recorded, with DBH measurements taken at approximately 1.3 m above ground level. The actual data collection situation is shown in Figure 5.



Figure 5. Display of the actual situation of partial plots in the Xiong'an dataset.

2.2.5. Dataset Creation

We first performed registration of the ground-based LiDAR point cloud data and UAV-derived point cloud data using CloudCompare. Subsequently, we conducted manual annotation for both datasets, classifying the sample plots into three categories: grassland, trees, and unclassified points. Each individual tree was assigned a unique label.

After completing the data annotation, we generated rectangular bounding boxes for each tree's terrestrial and UAV-derived point clouds and determined the tree centers based on these bounding boxes. Each terrestrial tree point cloud was then matched to its nearest neighboring UAV-derived tree point cloud. To establish tree correspondences,

we calculated the overlap ratio between the bounding boxes of each terrestrial tree and its nearest UAV-derived tree. Pairs with the highest overlap ratio exceeding 50% were identified as the same tree. In this process, the UAV-derived point cloud was designated as the input point cloud, while the terrestrial point cloud served as the ground truth.

After matching the UAV-derived tree point clouds with the terrestrial tree point clouds, we further associated them with field-measured tree data to determine the DBH of each tree. Specifically, a polygonal bounding box was generated for each UAV-derived tree. If a field-measured tree was located within the bounding box and was closest to its center point, it was identified as the corresponding field-measured tree for that UAV-derived point cloud. Through this two-step matching process, we established the DBH, terrestrial point cloud, and UAV-derived point cloud for each tree. Finally, we conducted a manual verification of the matching results to ensure accuracy.

In the end, we successfully matched 430 trees, with about 80% (351 trees) used as the training set and the remaining 20% (79 trees) as the test set. In accordance with the data processing method used in the FOR-instance dataset, the trees in the test set were obtained through stratified random sampling within the 13 sample plots. For each sample in the training set, we performed data augmentation, including operations such as translation and random rotation around the central axis. After augmentation, the training set contained 2808 input samples. No data augmentation was applied to the test set samples. Figure 6 shows the visualization of some of the point clouds from our training dataset. And the final dataset composition per region is detailed in Table 2.

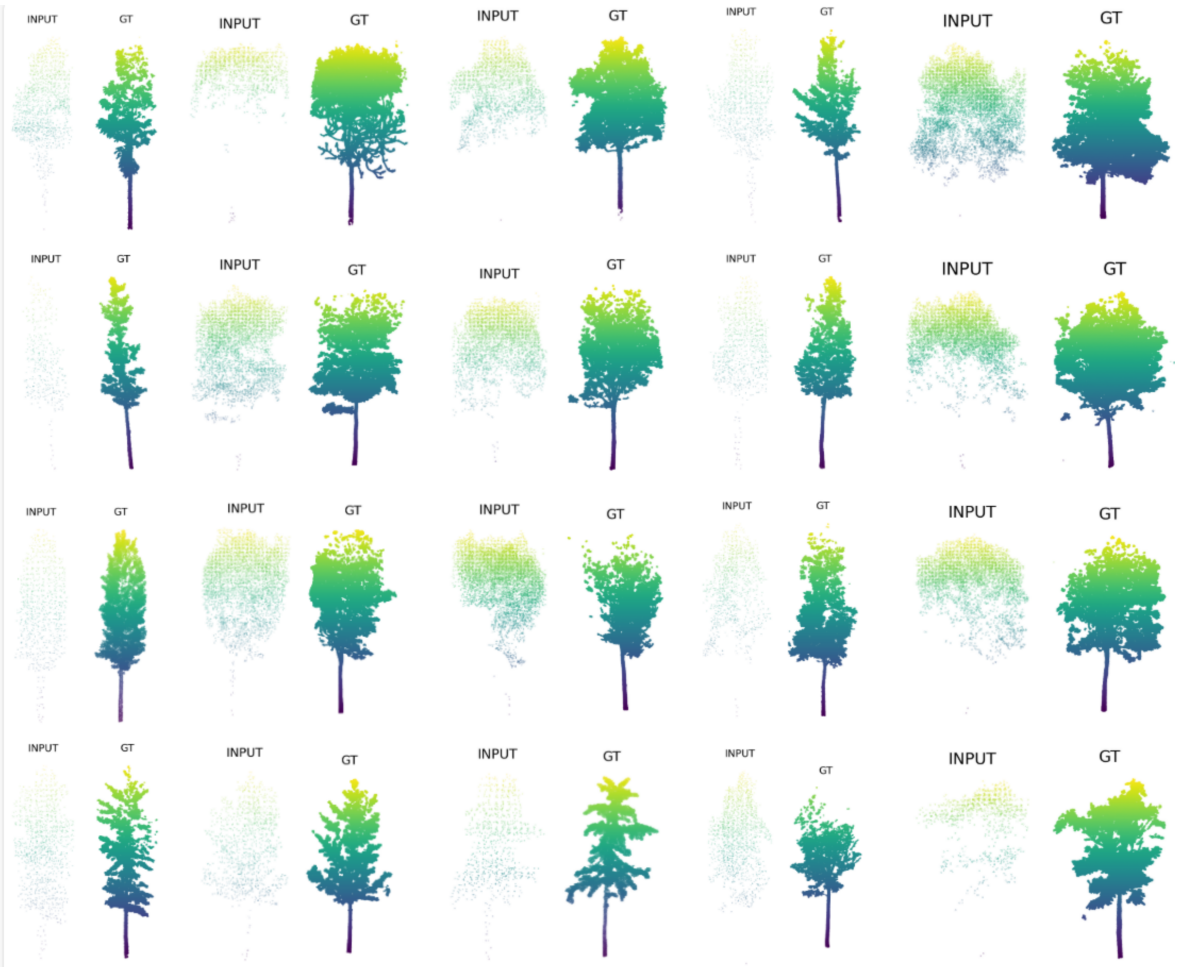


Figure 6. Partial visualization of the completed dataset from the Xiong'an dataset creation. The image assigns colors to the trees based on their height.

Table 2. Training–testing split statistics of the Xiong'an point cloud completion dataset. In the "Training Set" column, the number before the multiplication sign represents the original number of trees before data augmentation. We applied an 8× data augmentation to the trees in the training set, resulting in a final training sample size that is eight times the original number. The number of trees in the test set remains unchanged and has not undergone data augmentation.

PlotID	TrainSet	TestSet
150	35 × 8 = 280	8
151	42 × 8 = 336	10
152	27 × 8 = 216	6
153	31 × 8 = 248	7
155	28 × 8 = 224	7
157	31 × 8 = 248	7
158	21 × 8 = 168	5
159	23 × 8 = 184	5
160	32 × 8 = 256	7
161	24 × 8 = 192	5
162	16 × 8 = 128	3
163	24 × 8 = 192	5
164	17 × 8 = 136	4
Total	351 × 8 = 2808	79

3. Methods

3.1. SeedFormer Model

Based on the performance of the completion model and the practical results of tree completion, we selected SeedFormer [37] as the foundational model for our strategy. SeedFormer is an advanced deep learning neural network for point cloud completion. As shown in Figure 7, SeedFormer primarily consists of an encoder, a seed generator, and a coarse-to-fine generation module.

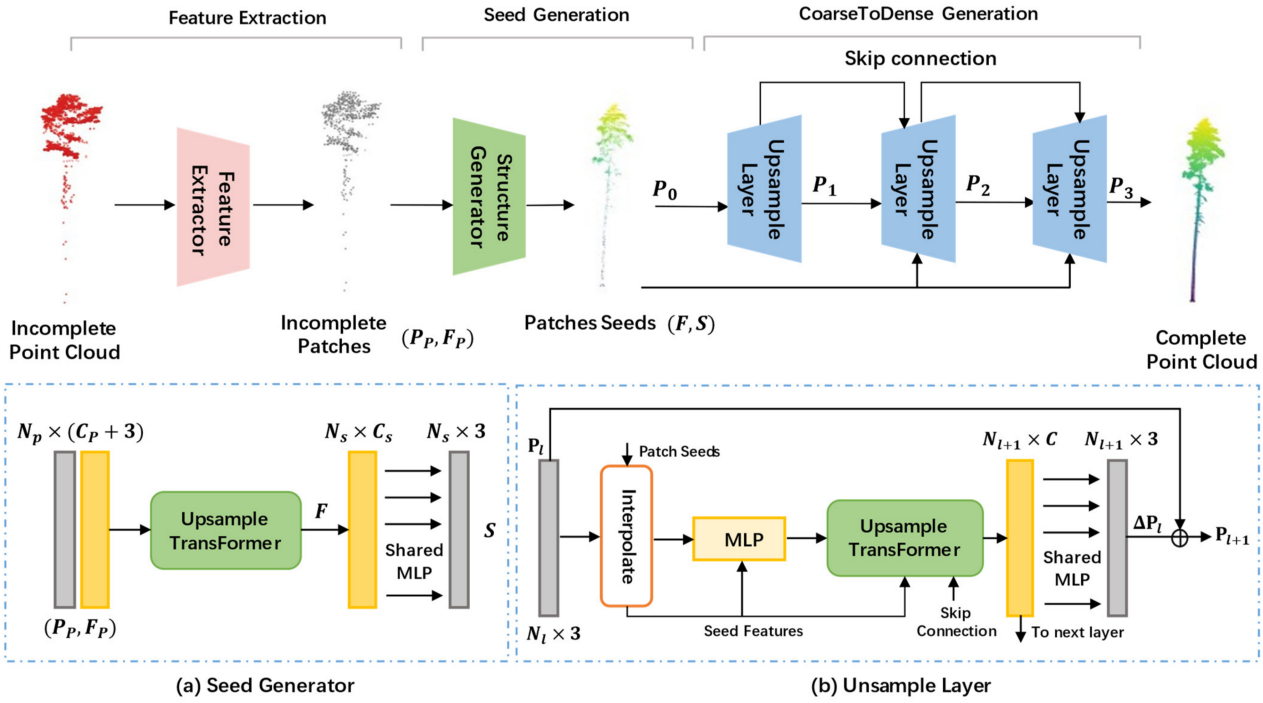


Figure 7. Structure of the SeedFormer model. (**a**) The seed generator is used to obtain Patch Seeds, which are then propagated to subsequent layers through interpolation of seed features. (**b**) In the coarse-to-fine generation process, multiple upsampling layers are used, with the Upsample Transformer applying skip connections and seed feature encoding to generate new points.

The encoder section uses Point Transformer and Set Abstraction layers to extract features from the input point cloud, outputting the center coordinates of Patch Seeds and the feature representations of the seed points. These coordinates and seed point features represent the local set characteristics of the input point cloud and provide support for subsequent point cloud completion. The seed generator aims to output a rough structure of the complete shape, specifically by generating seed points. It first utilizes a Transformer structure to transform the Patch Seeds features F_p into new seed features F:

$$F = UpTrans(F_p,\ P_p) \tag{1}$$

Subsequently, a multilayer perceptron (MLP) is used to map the seed features F into the three-dimensional coordinate space, yielding N_s seed points S:

$$S = \{x_i\}_{i=1}^{N_s} \in \mathbb{R}^{N_s \times 3} \tag{2}$$

During the seed generation phase, the Softmax normalization in the Transformer structure is removed to prevent the distribution of seed points from being constrained, allowing for a more flexible coverage of the complete shape. This design enables SeedFormer to both restore existing structures and reasonably infer the missing parts. The combination of the seed points S and features F forms the Patch Seeds, which serve as the foundation for the point cloud completion. Finally, the Upsample Transformer progressively generates the point cloud layer by layer, recovering the missing parts and ultimately producing the complete point cloud shape. The core idea of the Upsample Transformer is local attention aggregation, where, during new point generation, the spatial and semantic information of neighboring points is integrated to ensure that the new points maintain consistent geometric relationships with the surrounding structure. The specific process is as follows: the feature f_i of each point cloud is mapped through a multilayer perceptron (MLP) into three vectors: query (q_i), key (k_j), and value (v_j), which are used to calculate the relationship between the current point and its neighboring points and to generate new points. The query vector q_i comes from the feature of the current point cloud, the key vector k_j comes from the features of the Patch Seeds, and the value vector v_j is used to generate the new points. The attention calculation is performed by examining the relationship between the query and key vectors, and the specific formula is as follows:

$$a_{ijm} = \alpha_m\big(\beta(q_i) - \gamma(k_j) + \delta\big), \tag{3}$$

In this, α_m, β, and γ are feature mapping functions used for the mapping of the query and key, while the positional encoding δ integrates the geometric relationships in the point cloud and the relationships between seed features to ensure that the generated new points account for both spatial positions and local patterns. The final point computation is then performed by aggregating neighborhood information through the weighted value vector, as expressed in the following formula:

$$h_{im} = \sum_{j \in \mathcal{N}(i)} a_{ijm} * \big(\psi(v_j) + \delta\big), \tag{4}$$

Here, h_{im} represents the aggregated point feature, ψ is a mapping function used to adjust the value vector features, and δ serves as the positional encoding, ensuring that the generated new points are positioned appropriately in space. Finally, the new points are predicted for their three-dimensional coordinates through a multilayer perceptron (MLP), completing the point cloud upsampling process. This process is refined step by step through multiple layers of Transformer structures, ultimately restoring the complete

point cloud. The generated points effectively utilize the spatial distribution and semantic information of the existing points, thereby efficiently recovering the missing parts.

SeedFormer introduces a novel generative strategy combining local information enhancement with Transformer for the point cloud completion task. It preserves local geometric features through the Patch Seeds structure, aggregates neighborhood features using the Upsample Transformer, and employs a step-by-step refinement upsampling strategy. This approach ensures that the generated point cloud achieves optimal performance in both overall structural integrity and local detail fidelity. In real-world point cloud environments, factors such as noise, outliers, and occlusion can significantly impact the completion performance. However, the SeedFormer model structure is highly effective in addressing these challenges and improving completion performance. It leverages the Transformer mechanism to simultaneously learn global and local features, which allows it to effectively distinguish between noise points and true points, reducing the interference of outliers on the completion quality. Moreover, the model combines hierarchical feature extraction with global contextual information for shape reasoning, thereby alleviating the point cloud missing problem caused by occlusion to a large extent. Experimental results show that SeedFormer outperforms existing point cloud completion methods on multiple datasets, fully validating its exceptional capability in detail restoration and global shape reconstruction.

3.2. Hierarchical Random Sampling Method

To effectively process low- to medium-density UAV point cloud data and generate complete structure point clouds for individual trees, this paper proposes a height-based hierarchical random sampling strategy. The specific process is illustrated in Algorithm 1. First, the height information (z-coordinate) of the input point cloud is extracted, and the median value is calculated as the hierarchical threshold. Based on this threshold, the point cloud is divided into two parts: the low-height point cloud (below 50% height) and the high-height point cloud (above 50% height). Next, the proportion of the low-height point cloud, denoted as p_{low} (i.e., the ratio of low-height points to the total number of points), is calculated.

When $p_\text{low} < 0.15$, to ensure the representativeness of the low-level point cloud, 15% of the points are randomly sampled from the low-level point cloud, and the remaining points are randomly selected from the high-level point cloud. If $p_\text{low} \geq 0.15$, random sampling is performed directly across the entire point cloud. To ensure the stability and completeness of the sampling, if the number of points in one category of the point cloud is insufficient to meet the required sample size, random supplementation is performed to ensure the final number of points is met. Finally, the sampled points from both the low-level and high-level point clouds are merged to form the complete sampling result.

This sampling strategy ensures that the low-level point cloud accounts for at least 15% of the total points. This proportion is derived from a statistical analysis of the labeled point cloud, where points below 50% height account for approximately 15% of the total. For consistency, both the labeled point cloud and the input point cloud are sampled using this method, resulting in the labeled point cloud being uniformly sampled to 16,384 points, while the input point cloud is sampled to 2048 points.

Algorithm 1 Adaptive Point Cloud Sampling (APCS)

Input:
$\boldsymbol{P} \in \mathbb{R}^{N\times 3}$: Input point cloud,
$N \in N^{+}$: Target sample size
$\gamma \in [0,1]$: Median ratio of lower-layer points (pre-computed)
Output:
$\boldsymbol{P}_{samples} \in \mathbb{R}^{N\times 3}$: Sampled point cloud
Function APCS(P, n, γ)

Extract z-coordinates: $z \leftarrow P[:,2]$
Compute height threhold: $z_{th} \leftarrow \frac{\min(z)+\max(z)}{2}$
Partition point cloud: $P_{low} \leftarrow \{p \in P | p_z \leq z_{th}\}$; $P_{high} \leftarrow \{p \in P | p_z > z_{th}\}$;
Calculate actual ratio: $\alpha \leftarrow \frac{|P_{low}|}{|P|}$
if $\alpha < \gamma$ then
    $n_{low} \leftarrow \lfloor \gamma n \rfloor$;
    $n_{high} \leftarrow n - n_{low}$;
    if $|P_{low}| \geq n_{low}$ then
        $S_{low} \leftarrow UniformSubample(P_{low}, n_{low})$;
    else
        $S_{low} \leftarrow OverSubample(P_{low}, n_{low})$;
    if $\left|P_{high}\right| \geq n_{high}$ then
        $S_{high} \leftarrow UniformSubample\left(P_{high}, n_{high}\right)$;
    else
        $S_{high} \leftarrow OverSubample\left(P_{high}, n_{high}\right)$;
    $P_{sampled} \leftarrow Concatenate\left(S_{low}, S_{high}\right)$;
else
    if $|P| \geq n$ then
        $P_{sampled} \leftarrow UniformSubample(P, n)$;
    else
        $P_{sampled} \leftarrow OverSubample(P, n)$;
return $P_{sampled}$

3.3. Loss Function

The loss function is used to measure the discrepancy between the output point cloud and the ground truth point cloud. In this study, we employ the Chamfer distance [38] (CD), which is commonly used in most point cloud completion methods, as the loss function. The CD computes the average shortest distance between the output point cloud S_1 and the ground truth point cloud S_2, and is defined by the following formula:

$$CD(S_1, S_2) = \frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2} \parallel x - y \parallel_2 + \frac{1}{|S_2|}\sum_{y\in S_2}\min_{x\in S_1} \parallel y - x \parallel_2 \quad (5)$$

The first term of this formula encourages the output point cloud to be as close as possible to the points in the ground truth point cloud, while the second term ensures that the output point cloud adequately covers the ground truth point cloud.

To further improve the accuracy of individual tree point cloud completion and the structural integrity, we proposed a loss function consisting of three components: d_1, d_2, and d_3, weighted by hyperparameters α and β. The formula for the loss function is as follows:

$$L\left(Y_{seed}, Y_{dense}, Y_{dense'}, Y_{gt}, Y_{gt'}\right) = d_1\left(Y_{seed}, \widetilde{Y}_{gt}\right) + \alpha d_2\left(Y_{dense}, Y_{gt}\right) + \beta d_3\left(Y_{dense'}, Y_{gt'}\right) \quad (6)$$

The first term d_1 represents the distance between the seed point cloud Y_{seed} and the downsampled label point cloud $\widetilde{Y}_{gt}$. The second term d_2 calculates the distance between the completed point cloud Y_{dnese} and the ground truth label point cloud Y_{gt}. The third term d_3 computes the distance between the completed point cloud's lower 25% height points $Y_{dense'}$ and the corresponding portion of the label point cloud $Y_{gt'}$. In addition to the common loss functions, we introduced d_3 to enhance the model's focus on the lower tree canopy during the completion process. In most trees, the lower 25% of their height is sufficient to cover the portion of the tree at or below breast height, which is critical for accurately reflecting important structural features of the tree.

3.4. Point Cloud-Based Diameter at Breast Height Measurement Method

For point cloud data, higher-quality point clouds are more conducive to accurately calculating the diameter at breast height (DBH). TLS data has higher point cloud density and uniform distribution, providing good data support for point cloud-based DBH calculation. In contrast, ULS data is affected by factors such as occlusion, resulting in lower data quality in the lower canopy, which compromises the reliability of DBH calculation when used directly. To address this issue, we apply a point cloud completion method to effectively restore the tree structure in UAV point clouds. However, due to the potential large areas of missing data in ULS, the complexity of tree completion is higher, and the completed point clouds may still fail to meet the requirements for DBH calculation. Therefore, during the calculation process, we introduce certain condition checks to ensure that only higher-quality data is used for DBH measurement, thus improving the reliability of the results.

This paper develops a DBH calculation method [42] based on trunk slicing, referring to existing methods. The calculation process is shown in Figure 8. The approach mainly includes the following steps:

(1) Point cloud slices are extracted from the trunk at a height between 1.25 m and 1.35 m. To address potential noise issues in the original data, denoising is performed based on the consistency of the point cloud normal vectors. Since the normal vectors of the trunk point cloud are relatively stable, while the normal vectors of noise points fluctuate significantly, this paper calculates the rate of change of the point cloud's normal vector direction and removes points with large gradient changes to improve data quality. Additionally, to ensure the reliability and stability of the slices, the number of valid points within a slice must be no less than 25. If the number of points is insufficient, the slice thickness is appropriately increased to ensure the data is adequate to support subsequent calculations.

(2) After the preprocessing of the sliced point cloud, it is projected onto the XOY plane, and its convex hull is calculated to generate a polygon representing the tree trunk boundary. To further optimize the boundary shape, Gaussian smoothing is applied to the boundary points to reduce the impact of local outliers on measurement accuracy, making the extracted tree trunk contour more stable.

(3) To improve the stability and robustness of DBH measurement, this paper randomly generates 10 sets of lines that vertically pass through the centroid of the point cloud's

outer circle and calculates the length of the intersection between the lines and the tree trunk boundary's convex hull.

(4) The quality is assessed by calculating the standard deviation and mean of the 10 measurement values. If the standard deviation is greater than the set threshold ($\sigma < 0.15 \times$ mean measurement value), the slice point cloud is considered unable to accurately represent the tree trunk, and the DBH measurement result for that tree is discarded. Otherwise, the average of the 10 measurement results is taken as the final DBH measurement value.

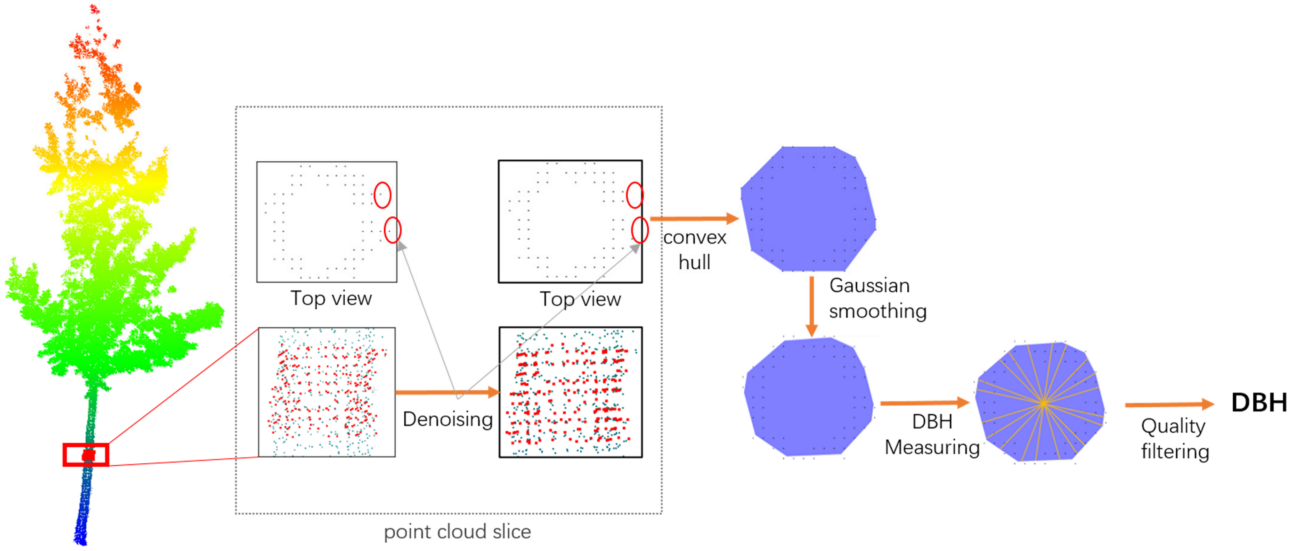


Figure 8. DBH measurement process. The blue points within the dashed box represent the point cloud slice of the tree trunk itself, while the red points represent the point cloud slice within the DBH range.

4. Results

4.1. Evaluation Metrics

Based on existing point cloud completion methods and commonly used evaluation metrics, this study selected L1 Chamfer distance (CD-L1), L2 Chamfer distance (CD-L2), and F1 score as the evaluation criteria for the point cloud completion performance. Specifically, CD-L1 and CD-L2 measure the difference between the completed point cloud and the ground truth point cloud. Smaller values indicate better completion performance by the network. The F1 score reflects the proportion of correctly reconstructed points in the completed point cloud; a higher score indicates better completion results. Furthermore, in conjunction with the point cloud-based diameter measurement method, the standard deviation between the completed and the actual tree diameter is quantitatively assessed. A smaller standard deviation suggests that the completed point cloud's diameter is closer to the actual measurement, indicating higher data consistency. This standard deviation provides an additional qualitative measure of the completion performance, particularly for low-to-medium density UAV point cloud data.

4.2. Experimental Setup

The hardware environment for the experiments in this study consists of an Intel(R) Xeon(R) E5-2650 v4 CPU, 128 GB of memory, and four 12 GB NVIDIA GeForce RTX 4090 GPUs; and one 24 GB NVIDIA RTX 4090 GPU; the operating system is Ubuntu 22.04. This overall hardware configuration provides a solid foundation for the experiments, ensuring the efficiency and stability of the training process. The experimental code was implemented

using the PyTorch (version 1.11.0 + cu113) deep learning framework. The model was trained for 150 epochs using the AdamW optimizer (version from PyTorch, developer: Meta AI, Menlo Park, California, USA), with an initial learning rate of 0.0001, which decays by half every 50 epochs. The batch size was set to 12.

4.3. Experient Results of Point Cloud Completion Models

4.3.1. Experiments on FOR-Instance Dataset

To validate the effectiveness of the proposed method, we conducted comparative experiments on five typical forest scenarios from the FOR-instance dataset: CULS, NIBIO, RMIT, SCION, and TEWIEN. SeedFormer was used as the baseline model, and performance comparisons were made by incorporating the improvements introduced in this study. Table 3 presents the quantitative evaluation results.

Table 3. Comparison of CD-L1 before and after combining SeedFormer with the proposed strategy on the FOR-instance tree completion dataset. The metrics used are L1 CD ×1000 (lower is better), L2 CD ×1000 (lower is better) and F-Score@1% (higher is better).

Region	SeedFormer			Ours		
	L1_CD	**L2_CD**	**FScore**	**L1_CD**	**L2_CD**	**FScore**
CULS	3.61	0.43	97.39	3.75	0.47	96.19
NIBIO	3.95	0.51	95.66	4.18	0.59	94.61
RMIT	10.32	2.97	58.47	9.74	3.06	62.91
SCION	4.23	0.56	94.95	4.40	0.63	93.94
TEWIEN	6.51	1.24	84.04	6.57	1.38	83.34
Average	5.72	1.15	86.10	5.73	1.23	86.20

Experimental results show that the overall completion performance of the method in this paper is comparable to SeedFormer, with the F-Score being almost the same (86.20% vs. 86.10%), but with slightly higher L1/L2-CD (<3%). This is mainly because the method in this paper focuses more on completing the lower canopy point clouds (e.g., the tree base), which are sparse in real point clouds. As a result, when calculating the symmetric CD, the completed point cloud cannot be fully covered by the real point cloud, slightly increasing the error value.

In different forest types, the method in this paper performs well in scenes such as coniferous forests (SCION) and deciduous forests (TEWIEN), optimizing the completion of lower-layer point clouds and improving physical space coverage. However, in the low-density point cloud RMIT scene, the completion performance of both methods significantly decreases. This may be due to the sparse original scanned point clouds and insufficient training samples, making it difficult for the model to learn complex occlusion patterns. This further highlights the significant impact of point cloud quality on completion performance.

In addition, Figure 9 shows that the method in this paper can better restore the lower trunk structure, effectively alleviating the collapse issue of SeedFormer in sparse areas. However, both methods still have limitations in restoring high noise and small-scale structures (such as fine branches). Future work could explore multi-scale feature fusion or physical prior constraints to improve the detail fidelity and overall stability of point cloud completion.

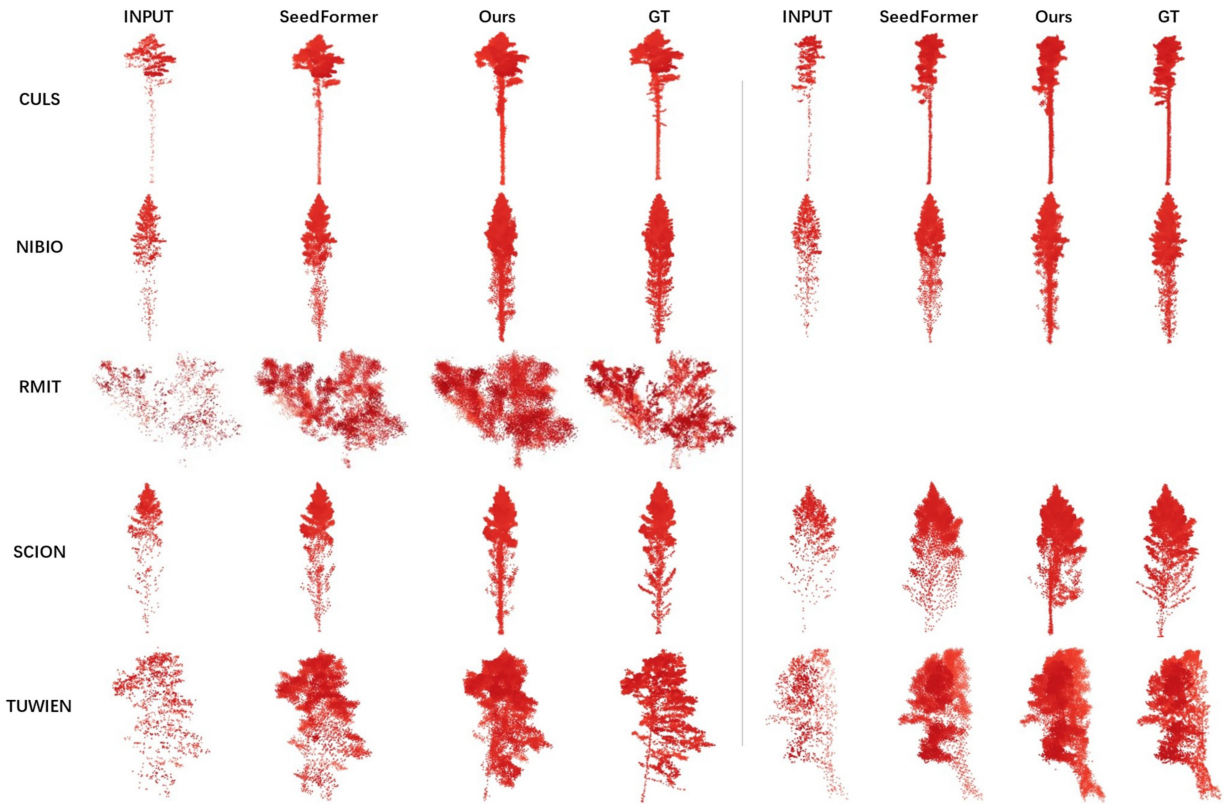


Figure 9. Model completion results based on the FOR-instance dataset. Since the RMIT region test set contains only one sample, only one result is displayed in the effect image.

4.3.2. Experiments on Xiong'an Dataset

We conducted a comparative experiment on 13 plots from the Xiong'an dataset. Using SeedFormer as the baseline model, we performed a performance comparison by incorporating the improved methods proposed in this study. Table 4 presents the quantitative evaluation results.

Table 4. Comparison of L1 CD before and after combining SeedFormer with the proposed strategy on the Xiong'an tree completion dataset. The metrics used are L1 CD ×1000 (lower is better), L2 CD ×1000 (lower is better) and F-Score@1% (higher is better).

PlotID	SeedFormer			Ours		
	L1_CD	L2_CD	FScore	L1_CD	L2_CD	FScore
150	5.49	0.60	93.53	5.36	0.57	94.06
151	6.93	1.32	82.75	6.90	1.39	82.28
152	16.12	5.45	30.36	14.00	4.50	49.97
153	12.42	3.86	53.71	11.45	3.21	59.31
155	11.90	2.82	70.37	11.40	2.55	73.92
157	9.99	2.02	77.86	10.06	2.28	72.94
158	7.37	1.04	84.18	7.43	0.99	85.01
159	8.37	1.52	79.29	8.13	1.46	80.32
160	8.27	1.32	82.23	8.44	1.33	82.23
161	12.08	3.30	57.05	10.45	2.29	74.77
162	9.48	1.83	78.96	9.02	1.55	79.55
163	10.38	2.09	77.33	9.04	1.51	78.02
164	11.34	2.70	71.85	11.32	2.63	72.36
Average	10.01	2.30	72.65	9.46	2.02	75.75

From the overall results, in the Xiong'an dataset, our method outperforms SeedFormer in the three evaluation metrics: L1_CD, L2_CD, and FScore-0.01. Specifically, L1_CD decreased from 10.01 to 9.46, L2_CD decreased from 2.30 to 2.02, and FScore-0.01 increased from 72.65% to 75.75%. This performance improvement may be attributed to the severe point cloud loss in the trunk region of the Xiong'an dataset, where our hierarchical random sampling strategy and loss function constraints effectively mitigated the issue of missing lower canopy point clouds, thereby enhancing the integrity and accuracy of the local geometric structure to some extent. It is worth noting that, under the same configuration, the completion performance on the FOR-instance dataset is generally superior to that on the Xiong'an dataset. This phenomenon can likely be explained by two factors: first, the more severe point cloud loss in the trunk region of the Xiong'an dataset, which limits the overall completion effect; second, the point clouds used in this experiment (drone point clouds) and the label point clouds (ground-based point clouds) are not from the same source, which may introduce some mismatching errors and affect the final completion results. The figure below shows the point cloud completion results of the two models in typical scenes. The visualization further validates the enhancing effect of the hierarchical sampling strategy on point cloud generation in sparse areas, and also demonstrates that, in the case of missing lower canopy point clouds, our method can still effectively recover key trunk structures, thus improving the overall integrity and quality of the point cloud.

4.3.3. Comparison of Sampling Methods Experiment

In the field of point cloud completion, the choice of sampling strategy significantly impacts the quality of the completion results. This experiment, based on the SeedFormer model, employs four different sampling strategies: Random Sampling (RS) [48], Furthest Point Sampling (FPS) [49], Adaptive Density Sampling (ADS) [50], and the Layered Random Sampling (LRS) proposed in this paper. Comparative experiments were conducted across multiple forest scenes in the FOR-instance dataset.

Table 5 shows that, despite using the same baseline model, different sampling strategies exhibit significant differences in point cloud detail representation. According to the data in the table, LRS (5.73) performs better than FPS (6.29) and ADS (5.92) in terms of L1 CD, second only to RS (5.71). This indicates that LRS can effectively reduce the distance error of the point cloud across the entire range. Although the F-Score of LRS (86.20%) is slightly lower than that of RS (86.54%), it is clearly superior to FPS (82.47%) and ADS (85.04%), suggesting that it has higher accuracy under the 1% threshold and is better at retaining key structural information of the point cloud.

Table 5. Comparison results of different sampling methods based on SeedFormer in various forest scenes of the FOR-instance dataset. The metrics used are L1 CD $\times 1000$ (lower is better), L2 CD $\times 1000$ (lower is better) and F-Score@1% (higher is better).

Method	CULS	NIBIO	RMIT	SCION	TUWIEN	L1_CD	L2_CD	FScore-0.01 (%)
RS	3.89	4.17	9.75	4.3	6.43	5.71	1.17	86.54
FPS	3.66	4.46	11.33	4.77	7.25	6.29	1.44	82.47
ADS	7.74	4.27	10.22	4.49	6.92	5.92	1.33	85.04
LRS (ours)	3.75	4.18	9.74	4.40	6.57	5.73	1.21	86.20

Combining Figure 10, it can be seen that the RS can better restore the tree contours. However, due to its random nature, in regions where the tree point cloud is sparse, RS may result in poor completion performance in low-density areas. This is because RS cannot effectively ensure an even distribution in sparse regions, which in turn affects the quality of point cloud completion. Through a combination of visual results and quantitative

analysis, it can be concluded that the LRS proposed in this paper significantly improves the completion quality of medium- and low-density point clouds in the For-instance dataset, demonstrating strong generalization ability across different forest scenes and providing stable completion results.

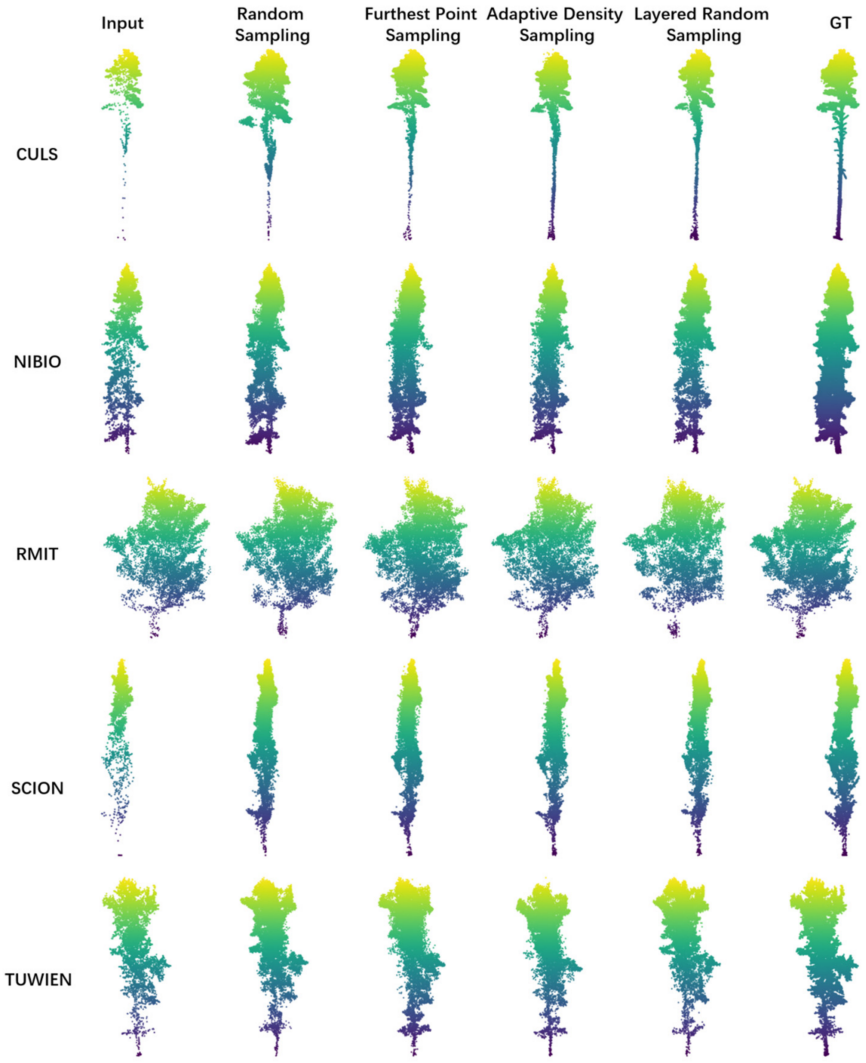


Figure 10. Completion results of different sampling methods based on SeedFormer in various forest scenes of the FOR-instance dataset. The image assigns colors to the trees based on their height.

In addition, compared to traditional RS methods, the LRS proposed in this paper does not introduce significant additional computational overhead. Although the method adds calculations of point cloud height information and layering operations, these operations only involve calculating the median and point cloud ratio, and their computational complexity is comparable to that of traditional methods. Therefore, the overall computational complexity does not increase significantly. The simple calculations in the data pre-processing stage do not affect the efficiency of the sampling process, and the random supplementation operations do not add extra computational burden, allowing the method to effectively maintain computational efficiency.

4.4. Experimental Evaluation of Point Cloud Completion in Improving the Reliability of DBH Measurement

4.4.1. Experiments on FOR-Instance Dataset

This experiment, based on five forest scenes from the FOR-instance dataset, compares the improvement effect of SeedFormer combined with our proposed method in the DBH measurement task. The evaluation metrics include root mean square error (RMSE) and the number of trees with undetected DBH (N_drop). The experimental results are shown in Table 6.

Table 6. RMSE between the predicted DBH values obtained from different point clouds and the reference values in the FOR-instance dataset (unit: cm). In the table, INPUT represents the input point cloud from the dataset, GT (ground truth) refers to the corresponding real tree point cloud, Completion denotes the completed model, and N_{drop} indicates the number of trees with undetected DBH. A smaller N_{drop} value signifies more stable DBH calculation results.

Region	Input		Completion (Seedformer)		Completion (Ours)		GT	
	RMSE	N_{drop}	**RMSE**	N_{drop}	**RMSE**	N_{drop}	**RMSE**	N_{drop}
CULS	3.56	35	3.84	19	3.03	15	3.12	16
NIBIO	5.55	327	5.41	251	5.31	214	5.44	216
RMIT	-	4	-	4	-	4	-	4
SCION	4.60	91	5.73	70	5.67	68	4.62	68
TEWIEN	14.59	57	14.03	45	12.70	44	11.47	43
Average	6.43	514	6.60	389	6.11	345	6.18	347

In the RMIT region, only four trees met the initial screening criteria. However, none of these trees satisfied the conditions during the DBH measurement process, making it impossible to compute a valid RMSE.

Currently, point cloud-based DBH measurement methods remain immature, making it challenging to ensure accurate DBH estimation for every tree. To evaluate the effectiveness of different methods, we introduce the N_{drop} metric, which represents the number of point clouds discarded due to unsuccessful DBH measurement. A lower N_{drop} value indicates that more trees have successfully obtained DBH measurements under the same sample size, thereby reflecting higher measurement stability.

Among all methods, the N_{drop} value for the input point cloud (Input) is the highest, likely due to the sparsity of the point cloud, which results in the lowest number of trees with a measurable DBH. After applying our method, the N_{drop} value is almost identical to that of GT, indicating that our approach effectively retains most of the useful data and performs well in reducing the number of trees with undetermined DBH.

Compared to SeedFormer, our method achieves a lower N_{drop} value, suggesting that the enhancement strategies improve the quality of the lower canopy region, making it more consistent with GT.

From the average results across all regions, the combination of SeedFormer with our enhancement method significantly reduces the N_{drop} value, while the RMSE is slightly lower than that of SeedFormer (6.11 vs. 6.43). This result indicates that our method effectively enhances the completion model's ability to recover sparse lower-layer point clouds of trees, thereby significantly increasing the number of trees with measurable DBH.

Figure 11 illustrates the scatter distribution of individual tree DBH predictions versus reference values for different point clouds.

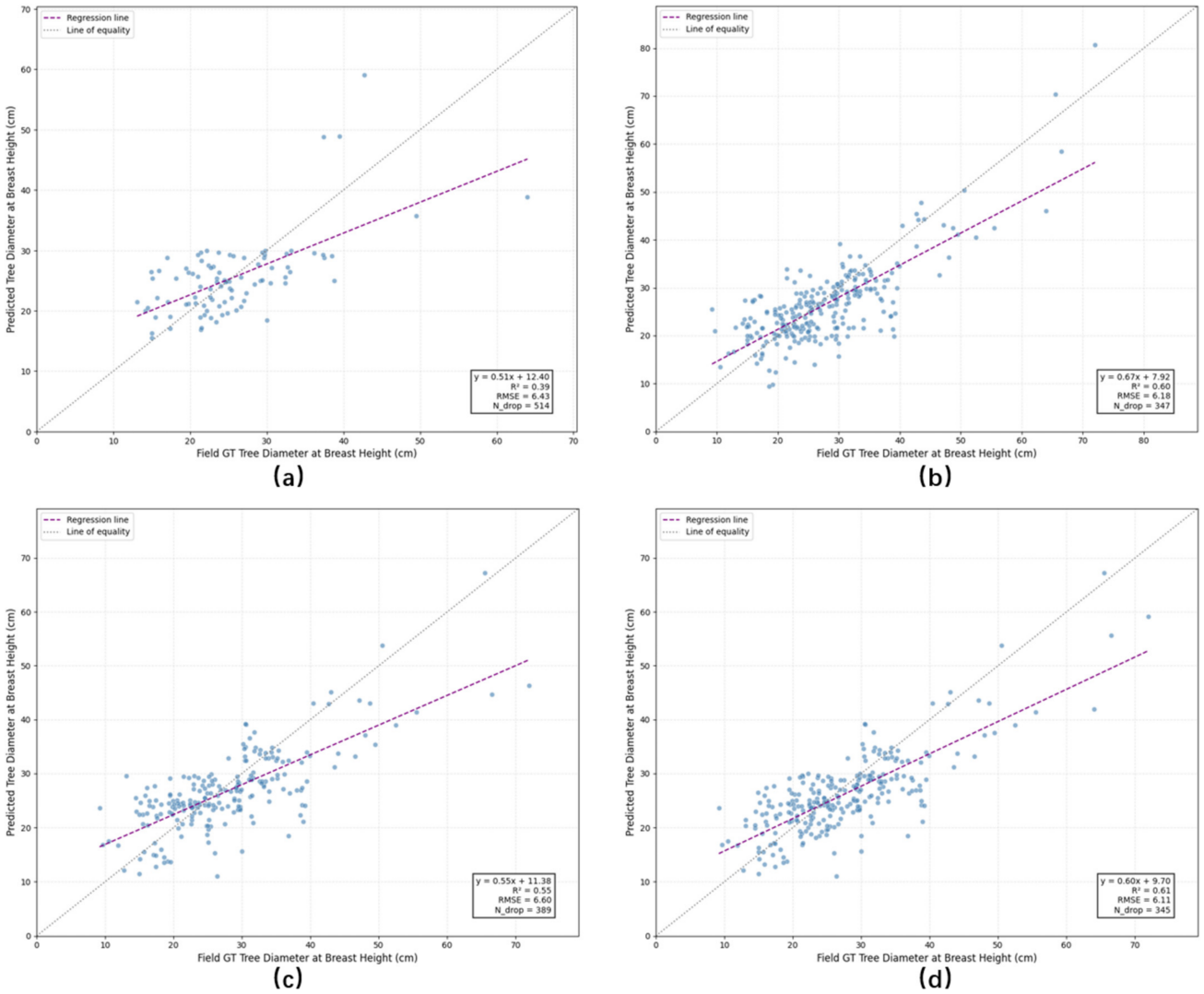


Figure 11. (**a**–**d**) Illustrate the scatter distribution of individual tree DBH predictions versus reference values across different point clouds in the FOR-instance dataset. (**a**) Shows the scatter plot of DBH predictions versus reference values obtained from sparse point clouds. (**b**) Presents the scatter plot using GT (ground truth) point clouds. (**c**) Depicts the scatter plot obtained from point clouds completed by SeedFormer. (**d**) Illustrates the scatter plot of DBH predictions based on point clouds enhanced with our proposed strategy on top of SeedFormer. N_{drop} in the figure represents the number of trees excluded due to the inability to measure DBH from the point clouds.

4.4.2. Experiments on Xiong'an Dataset

This experiment, based on 13 sample plots in the Xiong'an region, evaluates the improvement in DBH measurement performance when combining SeedFormer with our proposed method. The evaluation metrics include RMSE and the number of trees with undetected DBH (N_{drop}). The experimental results are shown in Table 7.

This experiment compares the performance of different methods in DBH measurement, evaluating them based on two metrics: RMSE and N_{drop}.

From the RMSE results, the measurement method based on the original point cloud (Input) achieves an RMSE of 3.64. However, due to missing and sparse input point cloud data, N_{drop} reaches as high as 407, indicating that a large number of trees cannot be effectively measured. In contrast, GT, serving as the reference value, has the lowest RMSE (1.22) and the smallest N_{drop} (51), demonstrating that a complete point cloud significantly improves DBH measurement accuracy and data usability.

Table 7. RMSE between predicted and reference DBH values for individual trees in the Xiong'an dataset, in cm. N_{drop} represents the number of trees with undetected DBH. A smaller N_{drop} indicates more stable DBH measurement results.

PlotID	Input		Completion (Seedformer)		Completion (Ours)		GT	
	RMSE	N_{drop}	RMSE	N_{drop}	RMSE	N_{drop}	RMSE	N_{drop}
150	3.46	39	5.53	16	5.55	11	1.63	7
151	3.62	40	5.54	7	5.14	3	1.29	1
152	-	33	5.57	12	6.2	7	1.24	4
153	-	38	4.31	3	2.84	3	0.93	0
155	-	35	3.58	24	4.12	24	1.26	2
157	-	38	3.14	12	2.99	10	1.07	4
158	5.06	23	6.38	4	4.37	1	1.05	1
159	-	28	4.39	18	4.34	12	1.35	8
160	3.17	37	4.75	8	4.77	3	1.19	2
161	1.17	27	8.49	21	5.68	17	1.75	17
162	-	19	4.62	9	3.18	6	1.69	5
163	-	29	5.35	8	4.42	0	0.67	0
164	-	21	6.15	15	6.64	8	0.68	0
Average	3.64	407	5.17	157	4.73	105	1.22	51

Among point cloud completion methods, the SeedFormer approach achieves an RMSE of 5.17, with N_{drop} reduced to 157, indicating that it enhances point cloud quality to some extent, enabling more trees to be measured. However, the relatively high RMSE suggests that structural deviations in the completed point cloud still contribute to measurement errors. When combined with our proposed enhancement method, the RMSE further decreases to 4.73, and N_{drop} drops to 105, outperforming SeedFormer on both metrics. This demonstrates that our approach not only reduces the number of undetected trees but also improves DBH prediction accuracy, bringing the measurement results closer to GT. Overall, our method exhibits better point cloud completion capability and greater stability in DBH measurement. Figure 12 illustrates the scatter distribution of individual tree DBH predictions versus reference values for different point clouds.

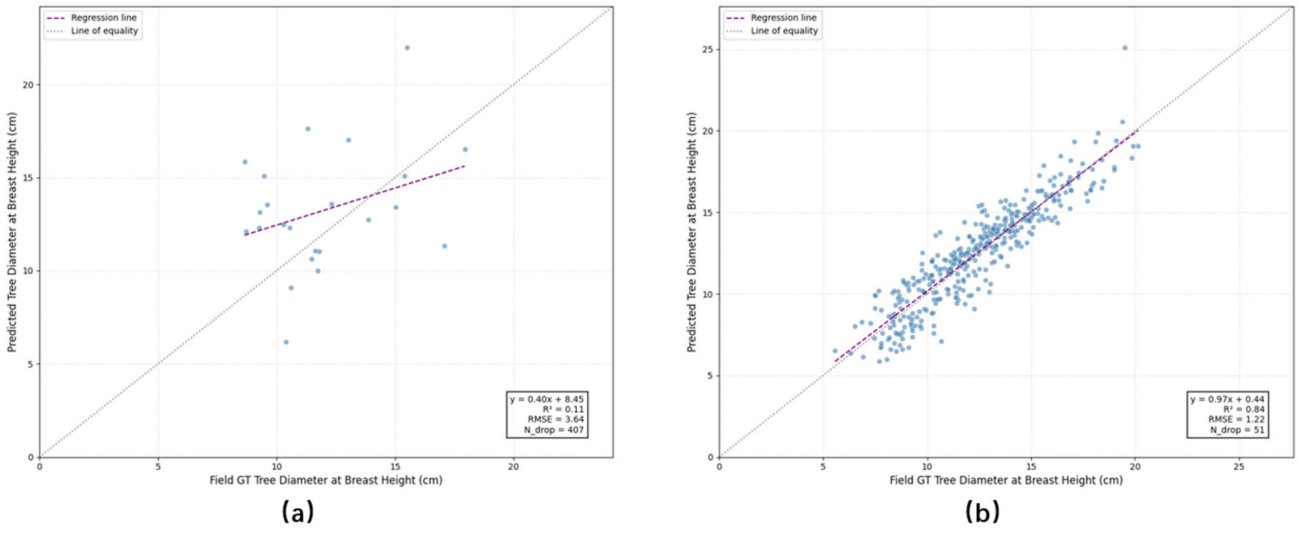


Figure 12. *Cont.*

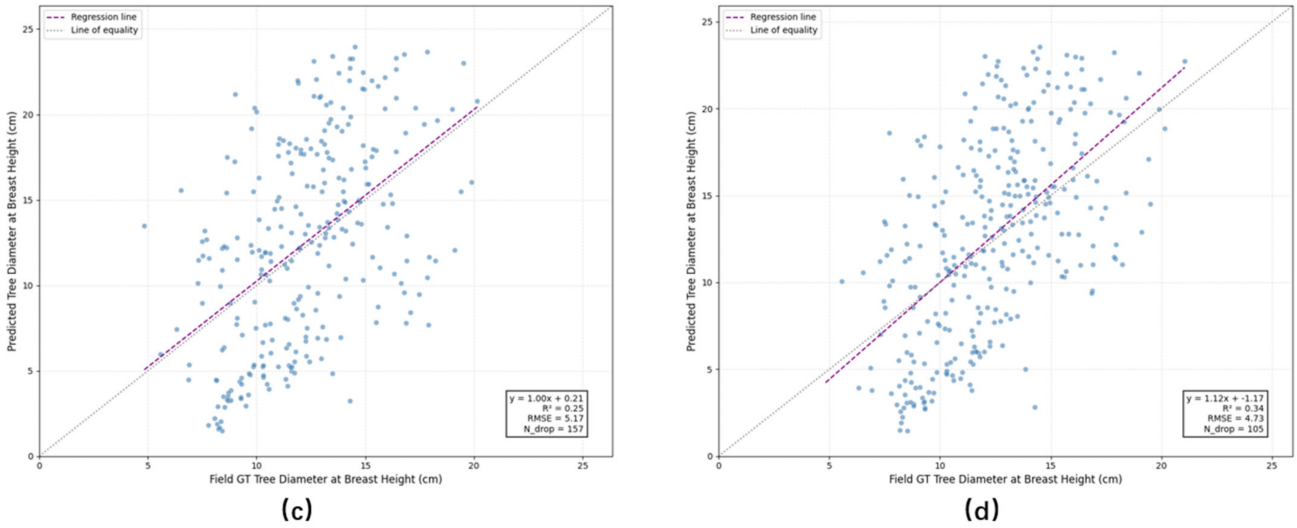


Figure 12. (**a**–**d**) Illustrate the scatter distribution of individual tree DBH predictions versus reference values across different point clouds in the Xiong'an dataset. (**a**) Shows the scatter plot of DBH predictions versus reference values obtained from sparse point clouds. (**b**) Presents the scatter plot using GT (Ground Truth) point clouds. (**c**) Depicts the scatter plot obtained from point clouds completed by SeedFormer. (**d**) Illustrates the scatter plot of DBH predictions based on point clouds enhanced with our proposed strategy on top of SeedFormer. N_{drop} in the figure represents the number of trees excluded due to the inability to measure DBH from the point clouds.

5. Discussion

5.1. Selection of Point Cloud Completion Models

In the selection of point cloud completion models, this study first compares and analyzes several representative point cloud completion models under optimal parameter settings, including PCN, GRNet, PoinTr, AdaPoinTr, SnowFlakeNet, and SeedFormer. Experiments were conducted on the FOR-instance dataset. The experimental results are shown in Table 8.

Table 8. Comparison of point cloud completion results using different models in various forest scenes of the FOR-instance dataset. The metrics used are L1 CD ×1000 (lower is better), L2 CD ×1000 (lower is better) and F-Score@1% (higher is better).

Method	CULS	NIBIO	RMIT	SCION	TUWIEN	L1_CD	L2_CD	FScore-0.01 (%)
PCN [29]	7.47	6.55	51.41	7.47	15.47	17.67	19.52	58.9
GRNet [18]	5.45	5.37	14.33	5.47	7.93	7.71	2.87	78.3
PoinTr [34]	5.01	4.67	13.11	5.02	8.37	7.24	2.08	76.03
AdaPoinTr [51]	5.77	5.14	23.47	5.67	9.78	9.97	6.26	72.33
SnowFlakeNet [36]	3.77	4.1	10.48	4.33	6.01	5.86	1.18	85.18
SeedFormer [37]	3.61	3.95	10.32	4.34	6.51	5.72	1.15	86.1

Based on the experimental results, PCN and GRNet are earlier point cloud completion methods. PCN uses a point-based encoder–decoder structure, while GRNet employs voxel-based global feature extraction. Both methods focus more on global structure during the completion process, but they have shortcomings in modeling local details, making it difficult to restore the complete branch and leaf structure, resulting in suboptimal overall completion performance. PoinTr and AdaPoinTr introduce the Transformer architecture, which more effectively models both global and local features of point clouds. In most scenarios, PoinTr outperforms GRNet in L1_CD and L2_CD (e.g., in the RMIT area, PoinTr's L1_CD is 13.11, while GRNet's is 14.33), indicating better performance in handling com-

plex point cloud structures. Notably, AdaPoinTr demonstrates significant advantages in completing complex, incomplete point clouds on public datasets (such as PCN and Completion3D) due to its adaptive query generation mechanism and denoising capabilities. The average CD error is reduced by approximately 12%, with higher training efficiency and generalization ability. However, the experimental results also show a deviation in AdaPoinTr's performance in completing medium- and low-density forestry point clouds compared to theoretical expectations. This could be due to a conflict between the denoising mechanism in AdaPoinTr and the sparse structures in the experimental data, resulting in some sparse points being mistakenly classified as noise during the denoising process. In particular, in medium- and low-density point cloud scenes collected by drones, the lower parts of the tree trunks were mistakenly filtered out due to sparse scanning, reducing the model's geometric accuracy and affecting the integrity of the tree structure. In contrast, SnowFlakeNet and SeedFormer significantly outperform other methods, especially in the restoration of tree crowns. Compared to SnowFlakeNet, SeedFormer also demonstrates better continuity in trunk restoration. Both methods, based on the Transformer structure, more accurately recover point cloud details through a layer-by-layer refinement approach and show stronger adaptability in complex forest environments, making the completed point clouds closer to the real distribution. For example, in the RMIT area, the L1_CD completion result of the PCN model is 51.41, while SeedFormer significantly reduces it to 10.32, showing its advantages in point cloud completion.

We have visually presented the completion results of the above-mentioned methods. From Figure 13, we can intuitively observe that PCN and GRNet only capture the basic shape of the trees, with the overall structure being quite vague. PoinTr and AdaPoinTr are able to generate basic tree shapes, but in some cases, the form is inaccurately captured, and their ability to recover branches and leaves is poor. SnowFlakeNet and SeedFormer clearly outperform the other methods, especially in the finer recovery of the tree canopy. SeedFormer also demonstrates better continuity in recovering the tree trunk. By comparison, we observe that the point cloud completed by SeedFormer is closest to the real point cloud, indicating its greater advantage in modeling tree topology and recovering local details. Ultimately, we selected SeedFormer as our baseline model, and when combined with the layered sampling and loss constraints proposed in this paper, it effectively improves the completeness of point cloud generation in the sparse lower areas of the trees, without significantly increasing geometric errors.

5.2. Analysis of Sampling Method Selection

In point cloud completion tasks, the sampling strategy has a significant impact on the spatial distribution characteristics of the input point cloud, which in turn affects the reconstruction performance of the completion network. In this study, we compared four different sampling strategies in the experiments: random sampling (RS), Furthest Point Sampling (FPS), Adaptive Density Sampling (ADS), and Layered Random Sampling (LRS), and analyzed the practical application of these sampling strategies.

Although the RS can uniformly cover the entire point cloud, it does not take into account the density distribution characteristics of the point cloud, which may result in insufficient sampling in low-density areas (such as tree trunks and fine branches), leading to significant information loss. When the completion network receives this uneven input, it may tend to fit the structure of high-density regions, while the completion performance in low-density areas is poorer. Therefore, in metrics such as L1 CD and F-Score, RS often shows better global matching performance, as the completed point cloud can align more accurately with the full point cloud in high-density areas, thus reducing global errors. However, in the sparsely populated parts of tree point clouds, RS struggles to provide

enough structural information, resulting in incomplete preservation of the overall structure of the tree.

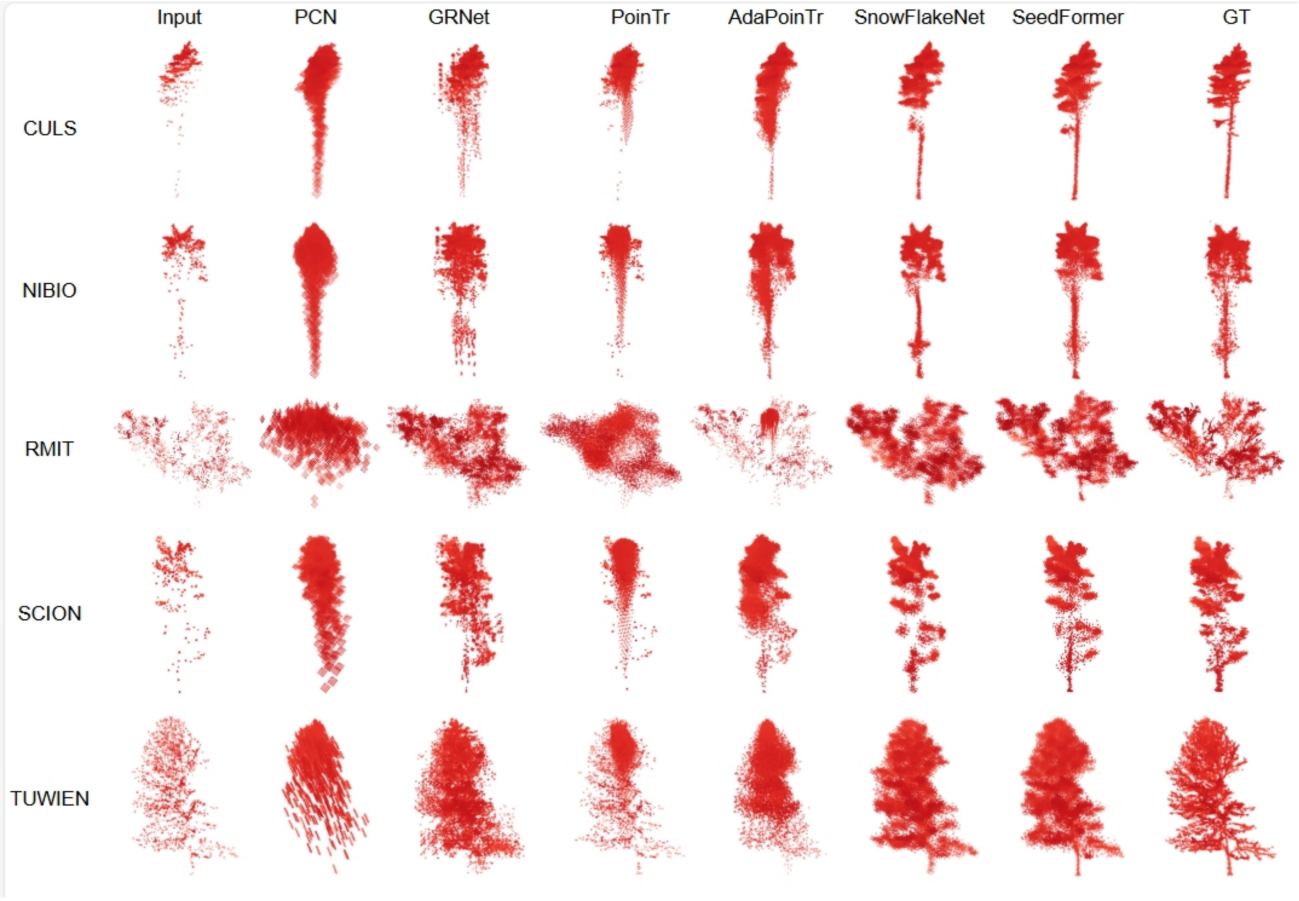


Figure 13. Completion results of different point cloud completion methods in various forest scenes of the FOR-instance dataset.

FPS is typically used for uniform point cloud sampling, achieving an even distribution in geometric space. However, for forest point clouds, especially UAV LiDAR data, the point cloud density in the canopy area is much higher than in the tree trunk area. When FPS samples in high-density regions, it tends to ignore the density characteristics of the point cloud, leading to insufficient sampling in low-density areas (such as the tree trunk), resulting in an incomplete trunk structure after completion. This is one of the reasons why FPS performs poorly in metrics like L1 Chamfer distance and F-Score.

ADS improves the completion effect by applying weighted sampling to the point cloud, giving more sampling points to low-density areas. From the visualization results, ADS achieves similar completion performance to LRS. However, compared to the LRS we propose, ADS cannot ensure the minimum proportion of low points in the point cloud, which may lead to incomplete tree trunks after completion. Additionally, since ADS primarily relies on point cloud density for adjustments, it is susceptible to local high-density outliers, causing over-sampling in certain areas, while important structures in truly low-density regions (such as the junction between the tree trunk and canopy) may still be ignored. In contrast, the LRS we propose can be seen as a special density-based sampling method that integrates the features of mid-to-low-density tree point clouds. By ensuring the proportion of low points while maintaining even coverage of the overall structure, this method is more suitable for different forest datasets. Through its layered strategy, it

compensates for the shortcomings of RS and FPS, providing better stability in point cloud completion in low-density areas compared to ADS.

In UAV point clouds, the canopy part is usually denser, while the trunk part is relatively sparse. When focusing on the overall structure of the tree for completion, the key lies in whether the sparse regions can obtain enough sampling points. ADS requires calculating the local density of the point cloud during sampling, especially in the canopy area, which may introduce additional computational overhead and increase processing time. In contrast, RS does not require extra computations, resulting in lower computational costs and allowing sampling to be completed in a very short time. In the specific scenario of mid-to-low-density UAV tree point clouds, our method achieves a completion effect similar to ADS in a shorter time, balancing both efficiency and completion quality:

5.3. Limitations and Outlook

5.3.1. Improvement of Sampling Methods

The hierarchical random sampling method proposed in this paper performs excellently in the completion task of low- to medium-density UAV point clouds, improving completion quality while ensuring computational efficiency. Compared to ADS, this method avoids the additional computational overhead caused by density calculation, enabling faster sampling and providing more stable completion results in low-density areas. However, this method still relies on the rationality of the hierarchical ratio, and its completion performance may exhibit some uncertainty in scenarios with significant differences in tree species or point cloud density distribution. Therefore, future research could combine the spatial distribution characteristics of point clouds to further optimize the hierarchical strategy, making the sampling method more adaptive and improving the generalization ability of the completion model.

5.3.2. Enrichment of Evaluation Metrics

In this study, we primarily use CD and F-Score as evaluation metrics for point cloud completion. These metrics can assess the geometric similarity between the completed point cloud and the ground truth point cloud from both global and local matching perspectives, but they do not directly evaluate the integrity, morphological consistency, or biological plausibility of tree structures. Ecological plausibility is an important factor that cannot be overlooked in forest point cloud completion tasks. To further enhance the comprehensiveness of the evaluation, future research could introduce more ecologically meaningful metrics. For example, the use of trunk taper error to measure whether it aligns with the gradual narrowing feature of real trees, or assessing the overall morphological consistency through trunk axis deviation.

5.3.3. Combined Usage Strategy of ULS and TLS Data

In large-scale and complex real-world forestry datasets (such as the Xiong'an dataset), the combined use of multi-source point cloud data remains a significant challenge. Although ULS data and TLS data are registered, directly using TLS data as labels for ULS data may still introduce substantial errors, affecting the generalization ability and transferability of the completion model. Therefore, how to reasonably integrate ULS and TLS data to fully leverage the wide-area coverage advantage of ULS and the high-precision characteristics of TLS remains an area for further exploration. Future research could employ deep learning-based multi-modal data fusion, spatial alignment optimization, or adaptive completion methods to reduce the discrepancies between different data sources and improve the applicability and robustness of point cloud completion models in complex forest environments.

6. Conclusions

This paper addresses the challenges of UAV point cloud data with low to medium density, particularly the issues of self-occlusion and sparse lower canopy point clouds. Two strategies are proposed to improve point cloud completion and maintain the structural integrity of trees. First, we perform stratification based on the statistical characteristics of the input point cloud and use a stratified random sampling method to obtain the point cloud. This ensures that the proportion of lower canopy points is maintained, which better preserves the original tree structure. Second, we incorporate the CD distance between the lower and middle layers of the point clouds into the loss function, further enhancing the model's focus on the lower canopy points and improving its ability to recover structure during the completion process. We validated our method on the FOR-instance and Xiong'an datasets. The results show that the introduction of stratified random sampling and the spatial constraint loss function significantly enhances the model's ability to recover the lower trunk structure. Although the model slightly underperforms the baseline model in terms of the symmetric Chamfer distance (CD) due to systematic errors in evaluation metrics, the visual results after applying our strategies are clearly superior to those of the original model. Additionally, we made improvements to the existing DBH measurement method and calculated the DBH on two datasets. DBH fitting was performed on the point clouds before and after completion, and the fitting results were compared with the actual measurements for analysis. Experimental results show that, after point cloud completion using the proposed strategy, the measured DBH accuracy is significantly better than the baseline model SeedFormer, further validating the effectiveness and application value of our method in tree structure completion.

Author Contributions: Conceptualization, Z.C.; methodology, Y.S.; software, Y.S.; validation, Y.S. and Z.C.; formal analysis, Y.S.; investigation, Y.S.; resources, Z.C.; data curation, Y.S. and X.X.; writing—original draft preparation, Y.S.; writing—review and editing, Z.C.; visualization, Y.S.; supervision, Z.C.; project administration, Z.C.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Xiong'an New Area Science and Technology Innovation. Special Project of Ministry of Science and Technology of China (2023XAGG0065).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Guo, Q.H.; Liu, J.; Tao, S.L.; Xue, B.; Li, L.; Xu, G.C.; Li, W.K.; Wu, F.F.; Li, Y.M.; Chen, L.H. Perspectives and prospects of Li-DAR in forest ecosystem monitoring and modeling. *Chin. Sci. Bull.* **2014**, *59*, 459–478.
2. Véga, C.; Renaud, J.; Durrieu, S.; Bouvier, M. On the interest of penetration depth, canopy area and volume metrics to improve Lidar-based models of forest parameters. *Remote Sens. Environ.* **2016**, *175*, 32–42. [CrossRef]
3. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. *Forests* **2021**, *12*, 131. [CrossRef]
4. Quan, Y.; Li, M.; Hao, Y.; Liu, J.; Wang, B. Tree species classification in a typical natural secondary forest using UAV-borne LiDAR and hyperspectral data. *GISci. Remote Sens.* **2023**, *60*, 2171706. [CrossRef]
5. Guo, Q.; Su, Y.; Hu, T.; Zhao, X.; Wu, F.; Li, Y.; Liu, J.; Chen, L.; Xu, G.; Lin, G. An integrated UAV-borne lidar system for 3D habitat mapping in three forest ecosystems across China. *Int. J. Remote Sens.* **2017**, *38*, 2954–2972. [CrossRef]
6. Xu, D.; Chen, G.; Jing, W. A single-tree point cloud completion approach of feature fusion for agricultural robots. *Electronics* **2023**, *12*, 1296. [CrossRef]
7. Ge, B.; Chen, S.; He, W.; Qiang, X.; Li, J.; Teng, G.; Huang, F. Tree Completion Net: A Novel Vegetation Point Clouds Completion Model Based on Deep Learning. *Remote Sens.* **2024**, *16*, 3763. [CrossRef]

8. Xu, H.; Huai, Y.; Zhao, X.; Meng, Q.; Nie, X.; Li, B.; Lu, H. SK-TreePCN: Skeleton-Embedded Transformer Model for Point Cloud Completion of Individual Trees from Simulated to Real Data. *Remote Sens.* **2025**, *17*, 656. [CrossRef]
9. Yurtseven, H.; Çoban, S.; Akgül, M.; Akay, A.O. Individual tree measurements in a planted woodland with terrestrial laser scanner. *Turk. J. Agric. For.* **2019**, *43*, 192–208. [CrossRef]
10. Xu, H.; Chen, W.; Liu, H. Single-wood DBH and tree height extraction using terrestrial laser scanning. *J. Forest Environ.* **2019**, *39*, 524–529.
11. Ko, B.; Park, S.; Park, H.; Lee, S. Measurement of tree height and diameter using terrestrial laser scanner in coniferous forests. *J. Environ. Sci. Int.* **2022**, *31*, 479–490. [CrossRef]
12. Liu, Y.; Wu, X.-Q. 3D shape completion via deep learning: A method survey. *J. Graph.* **2023**, *44*, 201–215.
13. Li, Y.; Wu, X.; Chrysathou, Y.; Sharf, A.; Cohen-Or, D.; Mitra, N.J. Globfit: Consistently fitting primitives by discovering global relations. In Proceedings of the ACM SIGGRAPH 2011, SIGGRAPH 2011, Vancouver, BC, Canada, 7–11 August 2011; pp. 1–12.
14. Kazhdan, M.; Hoppe, H. Screened poisson surface reconstruction. *ACM Trans. Graph. (TOG)* **2013**, *32*, 29. [CrossRef]
15. Nan, L.; Xie, K.; Sharf, A. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph. (TOG)* **2012**, *31*, 137. [CrossRef]
16. Kim, J.; Kwon, H.; Yang, Y.; Yoon, K. Learning Point Cloud Completion without Complete Point Clouds: A Pose-Aware Approach. In Proceedings of the 2023 International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 14157–14167.
17. Li, Y.; Dai, A.; Guibas, L.; Nießner, M. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2015; pp. 435–446.
18. Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; Sun, W. Grnet: Gridding residual network for dense point cloud completion. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 365–381.
19. Wang, X.; Ang, M.H.; Lee, G.H. Voxel-based network for shape completion by leveraging edge generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 13189–13198.
20. Yi, L.; Gong, B.; Funkhouser, T. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 15363–15373.
21. Hu, T.; Han, Z.; Shrivastava, A.; Zwicker, M. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
22. Hu, T.; Han, Z.; Zwicker, M. 3D shape completion with multi-view consistent inference. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 10997–11004.
23. Tang, J.; Han, X.; Pan, J.; Jia, K.; Tong, X. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4541–4550.
24. Li, Y.; Yu, Z.; Choy, C.; Xiao, C.; Alvarez, J.M.; Fidler, S.; Feng, C.; Anandkumar, A. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9087–9098.
25. Wen, X.; Xiang, P.; Han, Z.; Cao, Y.; Wan, P.; Zheng, W.; Liu, Y. Pmp-net: Point cloud completion by learning multi-step point moving paths. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7443–7452.
26. Li, S.; Gao, P.; Tan, X.; Wei, M. Proxyformer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9466–9475.
27. Rong, Y.; Zhou, H.; Yuan, L.; Mei, C.; Wang, J.; Lu, T. Cra-pcn: Point cloud completion with intra-and inter-level cross-resolution transformers. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; pp. 4676–4685.
28. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
29. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. Pcn: Point completion network. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 728–737.
30. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 206–215.
31. Wen, X.; Li, T.; Han, Z.; Liu, Y. Point cloud completion by skip-attention network with hierarchical folding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1939–1948.

32. Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; Le, X. Pf-net: Point fractal network for 3d point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7662–7670.
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, A.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
34. Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; Zhou, J. Pointr: Diverse point cloud completion with geometry-aware trans-formers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 12498–12507.
35. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.
36. Xiang, P.; Wen, X.; Liu, Y.; Cao, Y.; Wan, P.; Zheng, W.; Han, Z. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5499–5509.
37. Zhou, H.; Cao, Y.; Chu, W.; Zhu, J.; Lu, T.; Tai, Y.; Wang, C. Seedformer: Patch seeds based point cloud completion with upsample transformer. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 416–432.
38. Fan, H.; Su, H.; Guibas, L.J. A point set generation network for 3d object reconstruction from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 605–613.
39. Jucker, T.; Caspersen, J.; Chave, J.; Antin, C.; Barbier, N.; Bongers, F.; Dalponte, M.; van Ewijk, K.Y.; Forrester, D.I.; Haeni, M. Allometric equations for integrating remote sensing imagery into forest monitoring programmes. *Glob. Change Biol.* **2017**, *23*, 177–190. [CrossRef]
40. Xu, J.; Su, M.; Sun, Y.; Pan, W.; Cui, H.; Jin, S.; Zhang, L.; Wang, P. Tree Crown Segmentation and Diameter at Breast Height Prediction Based on BlendMask in Unmanned Aerial Vehicle Imagery. *Remote Sens.* **2024**, *16*, 368. [CrossRef]
41. Fakhri, A.; Latifi, H.; Samani, K.M.; Fassnacht, F.E. CaR3DMIC: A novel method for evaluating UAV-derived 3D forest models by tree features. *ISPRS J. Photogramm. Remote Sens.* **2024**, *208*, 279–295. [CrossRef]
42. Bogdanovich, E.; Perez-Priego, O.; El-Madany, T.S.; Guderle, M.; Pacheco-Labrador, J.; Levick, S.R.; Moreno, G.; Carrara, A.; Martín, M.P.; Migliavacca, M. Using terrestrial laser scanning for characterizing tree structural parameters and their changes under different management in a Mediterranean open woodland. *For. Ecol. Manag.* **2021**, *486*, 118945. [CrossRef]
43. Puliti, S.; Pearse, G.; Surový, P.; Wallace, L.; Hollaus, M.; Wielgosz, M.; Astrup, R. For-instance: A uav laser scanning benchmark dataset for semantic and instance segmentation of individual trees. *arXiv* **2023**, arXiv:2309.01279.
44. Puliti, S.; McLean, J.P.; Cattaneo, N.; Fischer, C.; Astrup, R. Tree height-growth trajectory estimation using uni-temporal UAV laser scanning data and deep learning. *Forestry* **2023**, *96*, 37–48. [CrossRef]
45. Kuželka, K.; Slavík, M.; Surový, P. Very high density point clouds from UAV laser scanning for automatic tree stem detection and direct diameter measurement. *Remote Sens.* **2020**, *12*, 1236. [CrossRef]
46. Wieser, M.; Mandlburger, G.; Hollaus, M.; Otepka, J.; Glira, P.; Pfeifer, N. A case study of UAS borne laser scanning for measurement of tree stem diameter. *Remote Sens.* **2017**, *9*, 1154. [CrossRef]
47. Girardeau-Montaut, D. *CloudCompare*; EDF R&D Telecom ParisTech: Paris, France, 2016.
48. Noor, S.; Tajik, O.; Golzar, J. Simple random sampling. *Int. J. Educ. Lang. Stud.* **2022**, *1*, 78–82.
49. Moenning, C.; Dodgson, N.A. *Fast Marching Farthest Point Sampling*; University of Cambridge, Computer Laboratory: Cambridge, UK, 2003.
50. Lin, Y.; Habib, A. An adaptive down-sampling strategy for efficient point cloud segmentation. In Proceedings of the ASPRS 2015 Annual Conference, Tampa, FL, USA, 4–8 May 2015.
51. Yu, X.; Rao, Y.; Wang, Z.; Lu, J.; Zhou, J. AdaPoinTr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14114–14130. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.