

Article

Diff-Tree: A Diffusion Model for Diversified Tree Point Cloud Generation with High Realism

Haifeng Xu ^{1,†}, Yongjian Huai ^{1,2,†}, Xiaoying Nie ^{1,2}, Qingkuo Meng ¹, Xun Zhao ³, Xuanda Pei ⁴ and Hao Lu ^{1,2,*}

¹ School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China; xuhaifeng@bjfu.edu.cn (H.X.); huaiyj@bjfu.edu.cn (Y.H.); niexy@bjfu.edu.cn (X.N.); mengqingkuo3110@bjfu.edu.cn (Q.M.)

² Engineering Research Center for Forestry-Oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China

³ State Forestry and Grassland Administration Key Laboratory of Forest Resources and Environmental Management, Beijing Forestry University, Beijing 100083, China; xunzhao@bjfu.edu.cn

⁴ Graduate School of Science, Tohoku University, 6-3 Aramaki Aza Aoba, Aoba, Sendai 980-8578, Miyagi, Japan; pei.xuanda.s4@dc.tohoku.ac.jp

* Correspondence: luhao@bjfu.edu.cn

† These authors contributed equally to this work.

Abstract: Three-dimensional (3D) virtual trees play a vital role in modern forestry research, enabling the visualization of forest structures and supporting diverse simulations, including radiation transfer, climate change impacts, and dynamic forest management. Current virtual tree modeling primarily relies on 3D point cloud reconstruction from field survey data, and this approach faces significant challenges in scalability and structural diversity representation, limiting its broader applications in ecological modeling of forests. To address these limitations, we propose Diff-Tree, a novel diffusion model-based framework for generating diverse and realistic tree point cloud with reduced dependence on real-world data. The framework incorporates an innovative tree realism-aware filtering mechanism to ensure the authenticity of generated data while maintaining structural diversity. We validated Diff-Tree using two distinct datasets: one comprising five tree species from different families and genera, and another containing five Eucalyptus species from the same genus, demonstrating the method's versatility across varying taxonomic levels. Quantitative evaluation shows that Diff-Tree successfully generates realistic tree point cloud while effectively enhancing structural diversity, achieving average MMD_{CD} and COV_{CD} values of (0.41, 65.78) and (0.56, 47.09) for the two datasets, respectively. The proposed method not only significantly reduces data acquisition costs but also provides a flexible, data-driven approach for virtual forest generation that adapts to diverse research requirements, offering a more efficient and practical solution for forestry research and ecological modeling.

Keywords: point cloud generation; diffusion model; tree structure; data driven



Academic Editor: Brenden E. McNeil

Received: 27 December 2024

Revised: 1 February 2025

Accepted: 4 March 2025

Published: 5 March 2025

Citation: Xu, H.; Huai, Y.; Nie, X.; Meng, Q.; Zhao, X.; Pei, X.; Lu, H.

Diff-Tree: A Diffusion Model for Diversified Tree Point Cloud

Generation with High Realism. *Remote Sens.* **2025**, *17*, 923. <https://doi.org/10.3390/rs17050923>

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

Three-dimensional (3D) virtual forest scene reconstruction is a key technology for revealing the complexity and dynamic changes of forest ecosystems. Research and applications such as forest radiation transmission, photosynthesis, ecological balance, and forest management and prediction all rely on a precise 3D forest structure [1–4]. A detailed forest model can be used to analyze forest growth, species interactions, and the impact of climate change on ecosystems, as well as provide scientific support for environmental monitoring

and resource management [5,6]. Despite the importance of Light Detection and Ranging (LiDAR) data in forest modeling, its data collection is often limited by specific regions, tree species, and sample scales, which impedes the implementation of large-scale and diversified scene modeling [7,8]. Therefore, studying the generation and expansion mechanism of tree point cloud data not only helps to address the shortcomings of existing data but also enhances the modeling accuracy and diversity of virtual forest scenes, providing new technical support for forest ecology, resource management, and environmental monitoring.

Currently, 3D forest scene modeling can be broadly divided into two categories: large-scale forest diversity scene modeling based on real data and virtual tree generation [9]. Forest scene modeling based on real data relies on various sensing technologies such as LiDAR and photogrammetry, which can provide accurate forest structure and spatial distribution information [10]. Virtual tree generation focuses on using computer graphics and artificial intelligence technologies to simulate the growth process and morphological characteristics of different tree species to construct diversified virtual forest scenes [8,9]. Both approaches employ key technologies such as voxel, point clouds, mesh, signed distance functions (SDF), neural radiance fields (NeRF), 3D Gaussian distributions, triplanes, and deep marching tetrahedra (DMTet), which enable accurate 3D representation and modeling of forests at different spatial scales [11].

Field surveys form the foundation of large-scale forest diversity scene modeling based on real data. These surveys collect information about the spatial distribution and structural information of different tree species in forests to reconstruct specific large-scale forest scenes [7]. However, the process of real data collection is complex, time-consuming, and labor-intensive. Although the advent of LiDAR significantly simplifies this process, point cloud data can be collected through LiDAR sensors mounted on different platforms, such as airborne LiDAR scanning (ALS), terrestrial LiDAR scanning (TLS), mobile LiDAR scanning (MLS), and unmanned aerial vehicle LiDAR scanning (ULS) [12–14]. Unfortunately, LiDAR data collection still faces certain limitations: the collection range is usually restricted to specific areas and fixed-scale samples, and the amount of data and tree species types obtained are still relatively limited. This directly impacts the diversity and generalization ability of virtual forest scenes at large spatial scales, making it difficult to meet the demand for large-scale, diversified forest scene modeling [6].

Virtual tree generation encompasses two main approaches: modeling and design [9]. The data used for modeling are generally point clouds or images, which are then used to construct mesh structures using computer graphics or deep learning algorithms. However, image data lack 3D spatial information and is not effective in inferring the structural parameters and morphological characteristics of trees [15,16]. In contrast, point cloud data cannot only depict the true shape of trees but also provide high-precision 3D spatial information, accurately capturing the geometric features and topological structure of trees. Tree structures can be reconstructed using point clouds collected from the real world, but the complexity of the data collection process makes it difficult to meet the requirements of large-scale data expansion and spatial diversification [13,17–19]. Virtual tree design methods, on the other hand, avoid reliance on real data and create tree models based on the structural parameters of target tree species, such as branch shapes, leaf tilt angle distributions, leaf area index, tree height, and diameter at breast height (DBH) [9]. However, this method requires a high level of understanding of tree structural parameters and faces significant challenges in designing diversified forest structures, often resulting in unnatural variations [20].

With the development of deep-learning technologies, diffusion models, as an advanced deep-learning method, are gradually transforming the generation and editing methods in fields such as image, text, and video [21]. Diffusion models based on point clouds may

provide a new solution to overcome the limitations in large-scale forest diversity scene modeling based on real data and virtual tree generation. By learning the distribution characteristics of existing point cloud data, new tree structures can be generated, effectively alleviating the problem of real data scarcity and generating diversified tree models similar to the distribution of original samples without relying on explicit rules. To date, there has been no research on tree point cloud diffusion methods. Existing diffusion models have been widely applied to regular objects with obvious continuity and symmetry characteristics (e.g., tables, chairs, airplanes); their application to generating tree point clouds, characterized by complex 3D forms and significant individual variations, remains unexplored [22–24].

The generation of realistic virtual trees necessitates robust objective evaluation and verification methods. Existing studies have proposed many vegetation modeling techniques, but their realism in generating vegetation models has not been precisely defined and verified [25]. Most existing evaluation methods rely on subjective judgment or comparison of images from multiple viewpoints, which cannot accurately reflect the realism of 3D shapes [26,27]. Recently, Polasek et al. [25] proposed an automatic no-reference realism predictor, ICTree, which includes a no-reference metric capable of validating the perceptual realism of vegetation models. ICTree also identifies certain significant vegetation features (such as branch angle, length, or volume) that significantly affect perceptual realism, as well as features (such as branch thickness or branch sequence deformation) with less impact. Although existing point cloud diffusion models can generate diversified samples, no clear method exists to evaluate the naturalness and perceptual realism of generated point clouds. While these advancements in diffusion models show promise, their application to tree point cloud generation presents unique challenges. Trees exhibit complex, non-uniform structures with significant variability between species and even individual specimens. This complexity necessitates specialized approaches that can capture the intricate branching patterns, leaf distributions, and overall architecture of trees while maintaining biological plausibility.

In response to the above issues, we propose Diff-Tree, a novel architecture specifically designed for generating a 3D tree point cloud. In the point cloud generation stage, we use a UNet-based diffusion structure, which, compared to existing point cloud diffusion methods, can generate point clouds more rapidly and achieve higher-quality results. Furthermore, to control the quality of the generated tree point cloud and ensure that the generated tree structures align more closely with realistic tree structure characteristics, we introduce a filtering mechanism for tree structures to select point clouds that better adhere to these characteristics. To validate the diversity and naturalness of Diff-Tree, we also perform reconstruction verification on the generated tree models and conduct a comparative analysis of tree structure parameters. The main contributions of this work are:

1. A new diffusion mechanism for tree point cloud to achieve diversified generation, which is the first diffusion model designed specifically for tree point cloud.
2. A perceptual realism filtering mechanism that ensures both diversity and natural appearance in generated tree point clouds.
3. Comprehensive validation of our model's performance using multi-source tree point cloud data, with detailed analysis of diversity and naturalness through visualization and reconstruction.

2. Related Work

2.1. Diffusion Models and 3D Generation Tasks

As 3D generation tasks become increasingly complex, traditional methods face significant limitations in terms of efficiency and scalability [28]. Diffusion models, based on the idea of progressively transforming data through stochastic processes, have emerged as

a powerful tool to tackle these challenges. These models have demonstrated remarkable potential in tasks such as image synthesis [29], denoising [30], and data restoration [31]. However, extending diffusion models from 2D images to 3D point clouds introduces two major challenges: (1) Point clouds are unordered and unevenly distributed, making feature extraction more difficult compared to 2D images, and (2) The increased dimensionality in 3D space adds significant computational complexity [22]. Despite these obstacles, diffusion models have shown promising results in addressing the complexity of 3D shape generation. Specifically, 3D diffusion models provide a robust framework for tasks requiring the synthesis, completion, or enhancement of 3D data [32,33].

Based on the above progress, our study focuses on generating 3D tree shapes—a domain still in its infancy. To inform our work, we reviewed the progress made in applying diffusion models to the generation of other 3D objects. The main variants of diffusion models include Denoising Diffusion Probabilistic Models (DDPM) [29], score-based generative models [34], and Stochastic Differential Equations (SDE) [35]. Compared with traditional generative models such as Variational Autoencoders (VAEs), diffusion models offer higher-quality outputs for 3D shape generation.

A notable example is the Latent Implicit Occupancy Network [36], which introduces a hierarchical latent space to surpass the generation performance of traditional denoising diffusion models (DDM). Latent Point Diffusion Model (LION) effectively tackles multimodal shape denoising and voxel-conditioned synthesis tasks, making it particularly suitable for 3D generation guided by textual descriptions and visual imagery. Furthermore, diffusion models excel in inferring depth, structure, and texture from one or more 2D views to generate accurate 3D reconstructions. For instance, Wang et al. [37] proposed the Score Jacobian Chain (SJC) method, enabling 3D generation using pre-trained 2D diffusion models without requiring additional 3D datasets. Xu et al. [38] further integrated the prior knowledge of diffusion models with reference images through Contrastive Language-Image Pretraining-guided (CLIP-guided) sampling strategies, achieving realistic 3D object reconstructions.

In addition to 3D shape generation, diffusion models enable the generation of 3D models or scenes from natural language descriptions, bridging the gap between textual information and 3D structures. Nichol et al. [39] were among the first to explore the use of DDPM for efficiently generating 3D point clouds from natural language. Subsequently, Chen et al. [40] and Wu et al. [41] introduced two-stage generation methods to improve the accuracy of outputs. Ren et al. [24] identified the complementary advantages of convolutional layers (from UNet) and attention mechanisms (from Transformers) and proposed a time-dependent dual-stream denoising model that balances visual quality and diversity. Meanwhile, to reduce runtime, Romanelis et al. [23] combined sparse voxel backbones with high-fidelity point branches to design a novel UNet-based diffusion architecture, significantly enhancing the efficiency of point cloud processing.

While most existing research focuses on conditional 3D generation tasks, our study aims to generate a realistic and diverse 3D tree point cloud without any specific input or condition (e.g., class labels, images, or text prompts). By capturing the underlying data distribution from noise, we explore the process of generating 3D tree structures without external guidance.

2.2. Virtual Tree Generation

As fundamental elements of virtual forests, 3D tree models can be created through various methods [42,43], which can be categorized into three main types: procedural modeling, sketch-based modeling, and interactive modeling.

Procedural modeling employs the L-system (Lindenmayer system) [44], a classic symbolic approach utilizing parallel string rewriting rules. This system simulates plant growth through iterative processes governed by grammar rules and symbol interpretations. The approach offers scalability and can be enhanced through optimization algorithms to improve refinement and generalization. In contrast, sketch-based modeling methods [43,45] provide a more intuitive and rapid means for users to generate structurally reasonable trees. However, these methods often encounter ambiguities when translating 2D sketches into 3D structures. Interactive modeling addresses this issue to some extent. For example, Liu et al. [46] introduced a virtual tree modeling system using six-degree-of-freedom (6DOF) motion controllers, enabling users to manipulate tree branches in 3D space with flexibility. Yuan and Huai [47] further combined this with Perlin noise to generate random twigs and leaves, enhancing the realism of the model.

In virtual tree modeling, the geometric form of vegetation conveys realism more intuitively than its physiological function. The most common input data types include RGB images and point clouds. For example, Cheng et al. [48] inferred tree structure from a single-depth image of a real tree and reconstructed the branches using cylindrical fitting. Similarly, Liu et al. [49] also conducted tree modeling based on a single image. However, these images consisted of user-drawn skeletons, which were further processed with a Generative Adversarial Network (GAN) to generate the tree structure. In addition to single-image modeling, multi-view methods also exist. For instance, Guo et al. [50] employed the binocular stereo vision method to obtain depth images, followed by the projection and fusion of all depth information to generate a dense point cloud, which was subsequently reconstructed. In contrast, point cloud-based methods are more intuitive and effective as they allow for the creation of tree skeletons and the capture of topological structure and key features [16,42,51].

Our study leverages point cloud data generated by diffusion models to capture and reproduce tree structural features, such as branch angles and lengths. Our ultimate goal is to produce novel 3D tree shapes with diverse structures and forms, thereby advancing beyond the limitations of traditional methods and providing an innovative approach to virtual tree modeling.

3. Materials and Methods

Our method consists of three main steps (Figure 1): point cloud diffusion, TreeIF filtering, and reconstruction with diversified validation. In Step 1, we first train the tree point cloud data using a diffusion model that leverages a UNet-based architecture. This model integrates a sparse voxel backbone with high-fidelity point branches to efficiently generate tree point cloud. Building on Step 1, we designed a filtering mechanism in Step 2 to further select tree point cloud data that align with realistic perception. This filtering mechanism evaluates the realism of the generated point clouds based on rendered images and 3D tree structural parameters, filtering out those that do not meet the standards. In Step 3, to validate the quality of the generated point clouds and the performance of Diff-Tree, we conducted 3D reconstruction and visualization to intuitively present the model's generation results. Furthermore, tree structural parameters were extracted to evaluate the realism and diversity of the generated point clouds. These three stages are detailed in Sections 3.1–3.3.

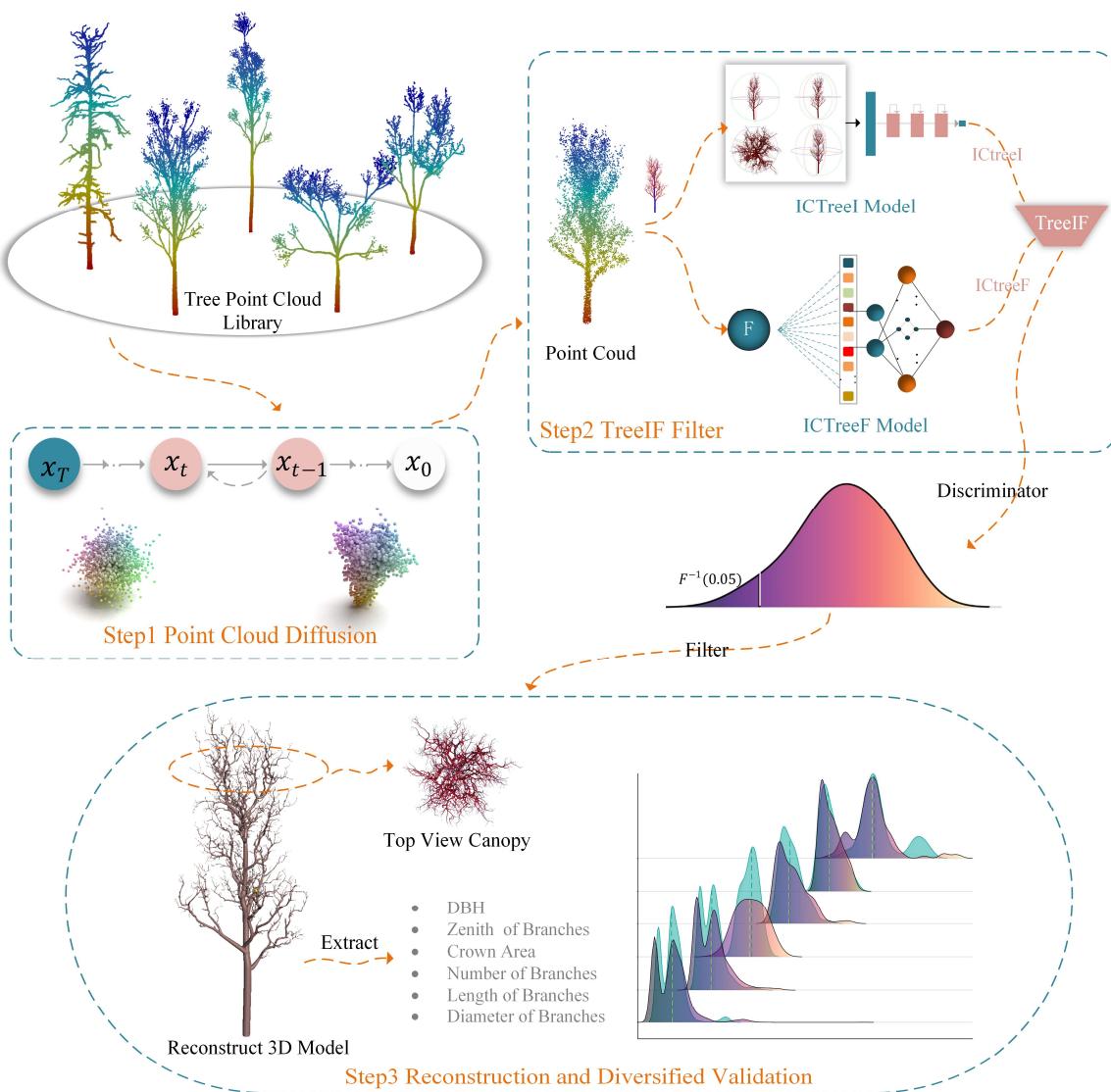


Figure 1. Overview of the tree point cloud generation process. The orange lines in the figure illustrate the complete workflow.

3.1. Diffusion Model for Point Cloud

The point cloud diffusion model, a specialized generative approach, operates through a sequential diffusion process. This process systematically adds Gaussian noise to original point clouds, gradually degrading their structural information. Subsequently, a reverse diffusion process reconstructs the original point cloud data from the noisy input. As shown in Figure 2, the diffusion model consists of two processes: forward and reverse. The purpose of the forward process is to perturb and transform original point clouds into noise, thereby constructing a latent space and providing training data for the reverse process. The reverse process aims to learn how to denoise and restore structures, ultimately generating realistic point clouds from noise. For the tree point cloud diffusion stage, we employ the sparse point-voxel diffusion (SPVD) model introduced by Romanelis et al. [23]. Furthermore, this SPVD model, based on a UNet diffusion architecture, combines a sparse voxel backbone with high-fidelity point branches to efficiently extract neighborhood information while preserving structural details, enabling the rapid generation of samples.

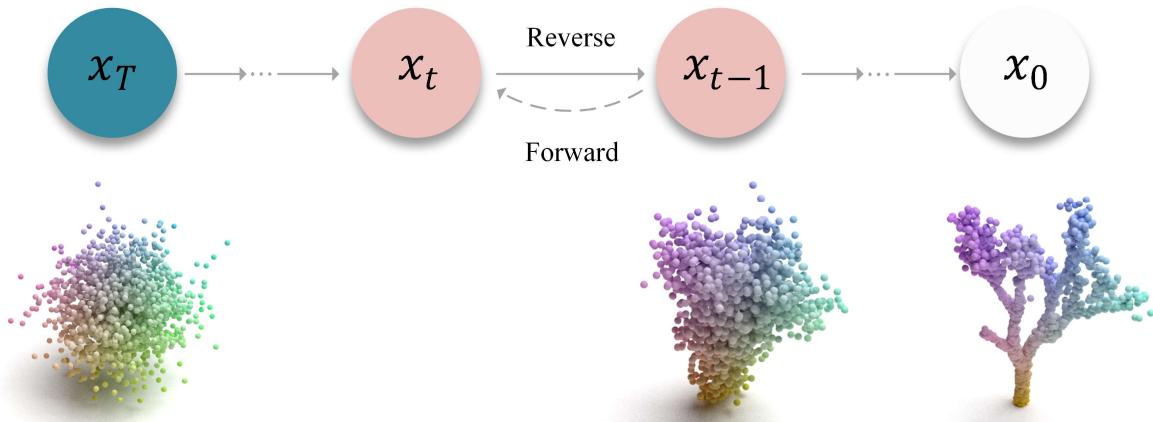


Figure 2. Visualization of the forward and reverse processes in the point cloud diffusion model. The forward process starts with $x_0 \in \mathbb{R}^{N \times 3}$ and progressively adds Gaussian noise to generate x_T , where x_T follows an isotropic Gaussian distribution. The reverse process then removes the Gaussian noise step by step to generate x_0 . The parameters for each step in the reverse process are predicted using a neural network model.

Forward: The forward diffusion process involves progressively adding noise to the original point cloud \mathbf{x}_0 , transforming it into Gaussian noise. Since $\mathbf{x}_0 = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^{N \times 3}$ as the original point cloud, where each $x_i \in \mathbb{R}^3$ represents a point in 3D space, the diffusion process progressively introduces noise to perturb the structure of the original point cloud. Assuming this process is described by a time parameter $t \in \{1, 2, \dots, T\}$, the point cloud data at time step t are defined as:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_{t-1} + \sqrt{1 - \bar{\alpha}_t} \mathbf{z}_t, \quad (1)$$

where $\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I})$ represents Gaussian noise. $\bar{\alpha}_t \in (0, 1)$ is a decay coefficient that controls the noise intensity. Specifically, $\alpha_t := 1 - \beta_t$, where $\beta_t \in \{\beta_1, \beta_2, \dots, \beta_T\}$ represents the predefined variance schedule, which determines the noise weight at each time step. Since the noise-adding process is defined as an approximate discrete-time Markov chain, $\bar{\alpha}_t$ is defined as follows:

$$\bar{\alpha}_t = \prod_{i=1}^T (1 - \beta_i). \quad (2)$$

As time steps advance, the original point cloud gradually transforms into Gaussian noise data \mathbf{x}_T , whose distribution are defined as follows:

$$\mathbf{x}_T \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_T} \mathbf{x}_0, (1 - \bar{\alpha}_T) \mathbf{I}\right). \quad (3)$$

Reverse: The reverse diffusion process corresponds to the actual generation phase, which is a Markov chain with Gaussian transitions. This process involves learning noise characteristics and progressively removing them. Specifically, it predicts the noise \mathbf{z} based on the current point cloud data \mathbf{x}_T and time step t , and progressively removes it. The reverse process can be updated as:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sqrt{\beta_t} \mathbf{z}, \quad (4)$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$. Here, $\epsilon_\theta(\mathbf{x}_t, t)$ is the noise predicted by a neural network, $\alpha_t := 1 - \beta_t$, and $\bar{\alpha}_t$ is defined in Equation (2). Furthermore, to accelerate

the generation process, the DDIM sampling rule [23,52] is introduced, which is defined as follows:

$$\mathbf{x}_t = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t, t). \quad (5)$$

During training, the model parameters are optimized by minimizing the error between the predicted noise $\epsilon_\theta(\mathbf{x}_t, t)$ and the true noise ϵ . Since the noise at time step t is computed in closed form, the objective function is solved by maximizing the variational lower bound, which is defined as follows:

$$\mathcal{L}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2], \epsilon \sim \mathcal{N}(0, \mathbf{I}). \quad (6)$$

The pseudocode for the training process and generation process is presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Training

Repeat until convergence:

1. Initialize $\mathbf{x}_0 \sim q(x_0)$
2. Set $t \sim Uniform(\{1, \dots, T\})$
3. Set $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
4. Perform gradient descent:

$$\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_T} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2$$

Algorithm 2 Generating

1. Initialize $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
 2. For $t = T$ to 1 do
 3. Set $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$
 4. # Update the reverse diffusion step:
 5. $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sqrt{\beta_t} \mathbf{z}$
 6. End for
 7. Return \mathbf{x}_0
-

Furthermore, to handle the complex branching structures and varying point densities characteristic of trees, the proposed approach employs and extends the Sparse Point-Voxel Diffusion (SPVD) architecture introduced by Romanelis et al. [23]. A sparse voxel backbone is introduced to efficiently capture local geometric features, while high-fidelity point-based branches preserve fine-grained details of the trunk and branches throughout the diffusion process. Integrating these two components enhances the process of feature extraction for tree point clouds without excessively increasing computational overhead.

Concretely, the model retains the DDPM-style U-Net [29] at its core while adapting it to sparse voxel operations: the raw point cloud is voxelized into a sparse grid, which is then passed through multiple Sparse Point Voxel Blocks (SPVB). Each SPVB contains residual convolutions, attention blocks, and up-/down-sampling layers specifically designed to process sparse data. This design is particularly advantageous for tree point clouds, as sparse voxel convolutions naturally group localized regions—such as dense trunks versus sparser branch areas—more efficiently than purely point-based or dense-voxel methods.

After extracting voxel-wise features, trilinear interpolation is employed to map those features back onto the original 3D point positions, merging them with features learned by a shared Multilayer Perceptron-based (MLP-based) point branch. Each point thus benefits

from both the neighborhood-aware information in the voxel domain and the fine-grained representations in the point domain. Additionally, temporal embeddings are projected into learnable scale–shift parameters via an MLP. These embeddings are then paired with sparse voxel features as graph nodes, leveraging PyTorch (version 2.0.0) Geometric to accommodate variations in voxel count across different tree samples efficiently. This design significantly improves network performance while supporting batch processing of point clouds with varying densities without modifying the architecture.

3.2. TreeIF Filtering Mechanism

The TreeIF Filtering Mechanism is designed to further improve the realism and diversity of generated point clouds by combining rendered images with 3D tree structural parameters for filtering. Specifically, TreeIF utilizes rendered 2D projection images and tree structural parameters (e.g., DBH and crown width) to comprehensively evaluate point clouds and select those that satisfy criteria for realism and structural consistency. This filtering mechanism effectively improves the quality of generated point clouds and ensures their usability in subsequent analyses. The filtering mechanism, as illustrated in Figure 3, comprises three components: tree perception metric calculation, TreeIF scoring, and dynamic threshold filtering.

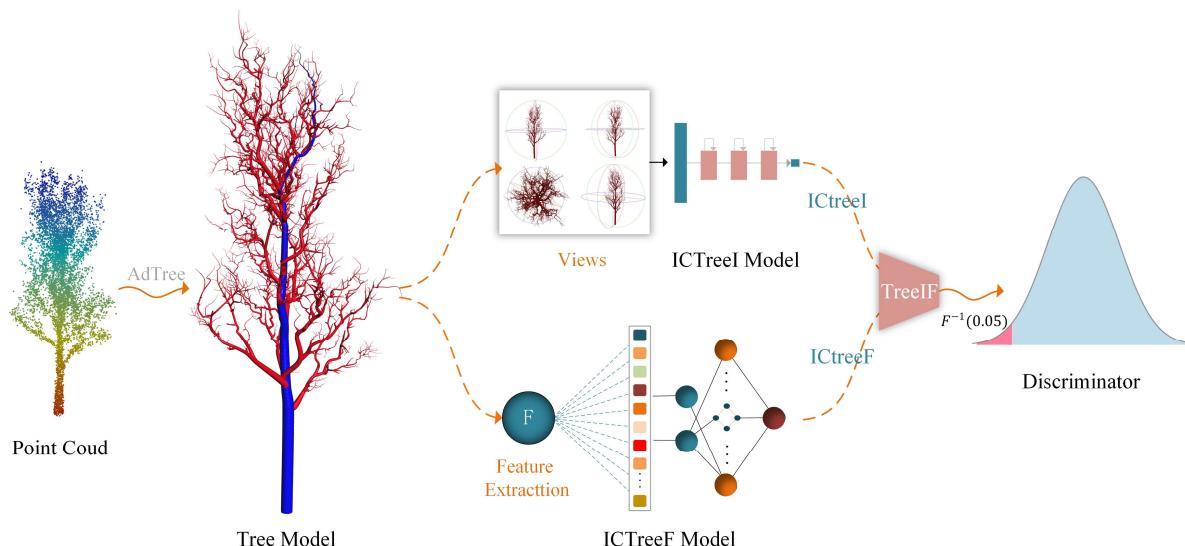


Figure 3. Filtering Mechanism for Tree point cloud. The filtering mechanism begins with the 3D reconstruction of tree point cloud, followed by the generation of 2D-rendered images and 3D structural parameter features. These features are used to compute the ICtreeI and ICtreeF metrics separately. The metrics are then combined using dynamic weights specific to each tree species to obtain the fused metric, TreeIF. Finally, dynamic thresholds are set for the discriminator based on the TreeIF distribution of each tree species.

Tree perception metric: We designed the TreeIF metric based on ICtreeI and ICtreeF, as proposed by Polasek et al. [25].

1. ICtreeI is a tree structure evaluation metric based on visual perception. As shown in Figure 3 (ICtreeI model), its input is a 2D image. It integrates rendering techniques and deep-learning prediction models. During the rendering process, grayscale shading and uniform lighting conditions are applied to avoid perception bias caused by environmental color factors. After rendering, ICtreeI extracts multiple tree views from a uniform height and combines these views with user perception scores to build a neural network based on Res2Net50, which predicts perception scores. Since user scores are positively correlated with realism, higher values indicate better results.

2. ICtreeF, as shown in Figure 3 (ICtreeF model), evaluates trees based on geometric features calculated from their 3D structure. To analyze and assess tree morphology, ICtreeF categorizes features into local and global features. Local features include tree skeleton topology and geometric statistics, such as branch width, length, angular variation, and branching direction. Global features encompass the entire tree model, including trunk length, leaf count, tree age, and volume footprint. In total, 29 features are used [25], and a DNN network serves as the regressor to predict the perceived realism of 3D tree models. Similar to ICtreeI, ICtreeF produces higher scores for models with better realism.

TreeIF scoring: We combined ICtreeI and ICtreeF into a single metric, TreeIF, using a weighted linear combination. Since ICtreeI and ICtreeF exhibit different distributions across tree species, a dynamic weight adjustment mechanism is applied to balance their contributions to TreeIF for each species. TreeIF is defined as follows:

$$\text{TreeIF}_i = \omega_{\text{ICtreeI}} \frac{\text{ICTreeI}_i}{\max(\text{ICTreeI})} + \omega_{\text{ICtreeF}} \frac{\text{ICTreeF}_i}{\max(\text{ICTreeF})}. \quad (7)$$

Here, ICTreeI_i and ICTreeF_i denote the ICtreeI and ICtreeF values for the i -th tree point cloud, respectively. ω_{ICtreeI} and ω_{ICtreeF} are the dynamic weights for ICtreeI and ICtreeF, satisfying $\omega_{\text{ICtreeI}} + \omega_{\text{ICtreeF}} = 1$. The dynamic weights are computed based on standard deviation to balance the influence of ICtreeI and ICtreeF on the fused metric across different distributions. The weight calculation formulas are as follows:

$$\omega_{\text{ICtreeI}} = \frac{\sigma_{\text{ICTreeF}}}{\sigma_{\text{ICTreeI}} + \sigma_{\text{ICTreeF}}}, \quad \omega_{\text{ICtreeF}} = \frac{\sigma_{\text{ICTreeI}}}{\sigma_{\text{ICTreeI}} + \sigma_{\text{ICTreeF}}}, \quad (8)$$

where $\sigma_{\text{ICTreeI}} = \sqrt{\frac{1}{N} \sum_i^n (\text{ICTreeI}_i - \mu_{\text{ICTreeI}})^2}$ and $\sigma_{\text{ICTreeF}} = \sqrt{\frac{1}{N} \sum_i^n (\text{ICTreeF}_i - \mu_{\text{ICTreeF}})^2}$ are the standard deviations of the two metrics, and μ_{ICTreeI} and μ_{ICTreeF} represent the mean values of ICtreeI and ICtreeF for each tree species.

Dynamic threshold filtering: By calculating the TreeIF for each tree, the distribution of TreeIF for each species can be obtained. Subsequently, we applied the fifth-percentile outlier detection method [53] to dynamically determine the filtering threshold for each species. The filtering threshold T_{TreeIF}^k for the k -th species is defined as:

$$T_{\text{TreeIF}}^k = F_{\text{TreeIF}}^{-1}(0.05). \quad (9)$$

Here, $F_{\text{TreeIF}}^{-1}(0.05)$ represents the fifth percentile of the TreeIF distribution. Finally, point clouds with $\text{TreeIF}_i \geq T_{\text{TreeIF}}^k$ are selected as qualified samples.

3.3. 3D Reconstruction and Diversity Validation

In the next stage, the generated tree point cloud was evaluated for structural consistency and distribution rationality through reconstruction and parameter extraction. The point clouds were first processed using Adtree [16] and TreeQSM [54] to conduct structural reconstruction and visualization, as shown in Figure 4. The ability of Adtree (version 1.1.2) and TreeQSM (version 2.4.1) to successfully reconstruct the point clouds indicates that the generated tree structures are usable and conform to natural tree forms. Subsequently, structural parameters were extracted, including DBH, Zenith Angle of Branches, Crown Area, Number of Branches, Length of Branches, and Diameter of Branches.

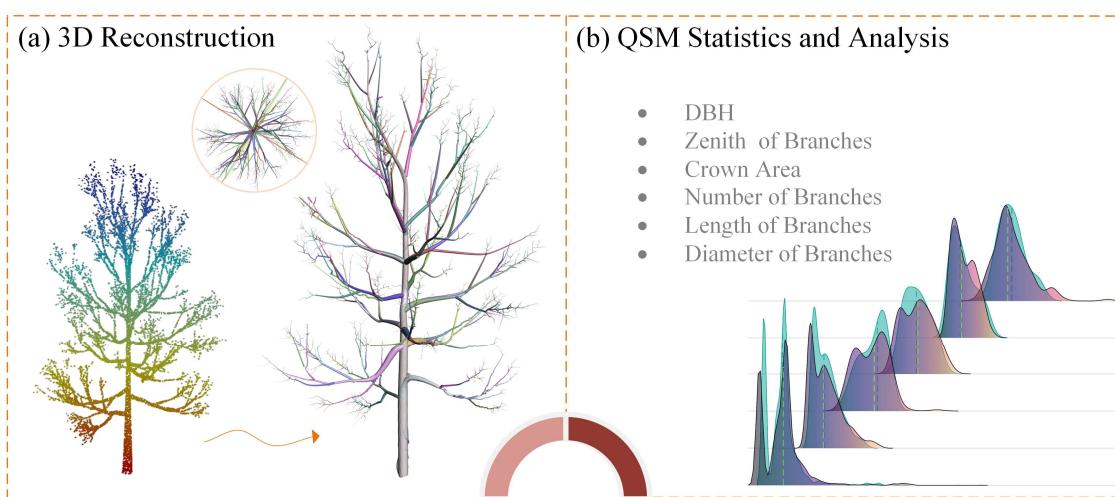


Figure 4. Comparison of point cloud reconstruction and structural parameters. **(a)** Visualization of the point cloud reconstruction results; **(b)** extracted structural parameters and their probability density distribution.

Specifically, the point cloud data were first denoised and segmented, dividing the point cloud into the stem and individual branches. During the topological reconstruction phase, the cover-sets method was used to partition the point cloud into small subsets, effectively segmenting the tree structure. Next, geometric reconstruction was conducted by fitting cylinders to represent the geometric features of each branch, including the branch's length, diameter, and volume. In this way, the DBH of the tree trunk and the diameter of each branch were extracted from the fitted cylinder parameters. The number of branches was determined by detecting and segmenting the branch points. Other parameters, such as the zenith angle of the branches and crown area, were also estimated using similar geometric fitting processes. These parameters were selected to comprehensively represent both the overall morphology and local structural features of the trees. For example, DBH serves as a critical indicator of trunk size, closely related to biomass estimation; the zenith angle of branches captures the spatial distribution of branch growth; crown area reflects canopy morphology and coverage; and branch number, length, and diameter further describe branching structure and detail characteristics [19,55]. It is worth noting that, during the diffusion process, tree height was not included as an evaluation parameter under the normalization framework, as it contributes limited additional structural insight when the dataset is scaled uniformly.

3.4. Dataset and Experimental Setup

Datasets: To validate the generalization capability of the Diff-Tree diffusion method across different tree species, we utilized two sets of experimental data.

- TD1: Tree Dataset 1 (TD1) was obtained from the tree point cloud dataset provided by Dobbs et al. [56]. This dataset included five distinct tree species from different genera: *Apple*, *Cherry*, *Ginkgo*, *Pine*, and *Walnut*, with 100 trees per species. The naming conventions for these tree species follow those used by Dobbs et al. [56], as the dataset does not specify the exact species within each genus.
- TD2: Tree Dataset 2 (TD2) comprised five eucalyptus species generated using OnyxTree software (version 7.0 developed by Onyx Computing Inc., Cambridge, MA, USA), specifically *Kombolgiensis*, *Paliformis*, *Pyrocarpa*, *Sieberi*, and *Yilgarnensis*, each containing 100 trees. Since the OnyxTree-generated data were in mesh format. We applied a

method of uniform sampling [57] on the grid surface to convert these mesh models into point cloud data.

The visualization of the two datasets, illustrated in Figure 5, revealed significant differences between TD1 and TD2. Specifically, the trunks in TD1 appeared relatively thicker, whereas the canopy structures in TD2 exhibited a higher degree of structural complexity. Since this study focuses on the diffusion of tree trunk and branch structures, both datasets excluded leaf structures. Furthermore, to ensure compatibility with the model's input requirements, all point clouds were downsampled to 15,000 points and normalized within a unit sphere [23]. During the training phase, both TD1 and TD2 were divided into 70% for training and 30% for testing.

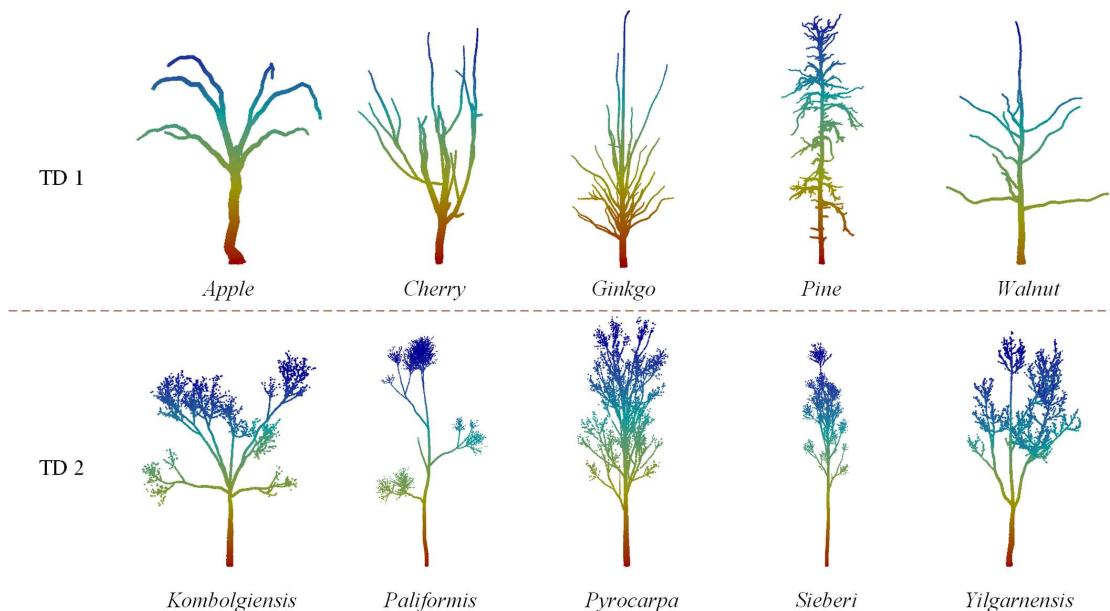


Figure 5. Visualization of the point clouds for TD1 and TD2. Both datasets were normalized and displayed within a unit sphere. It can be observed that the trunk thickness and canopy complexity exhibit noticeable differences between TD1 and TD2.

Evaluation Metrics: To enable a comprehensive comparison, we followed previous studies [22–24] and employed Chamfer Distance (CD) and Earth Mover's Distance (EMD) as the evaluation metrics for generated samples. CD quantifies the difference between two point sets by measuring the distance between each point in one set and its nearest neighbor in the other set. With two point sets X and Y , each containing the same number of points, it is defined as follows:

$$CD_{(X,Y)} = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} \|x - y\| + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} \|y - x\|. \quad (10)$$

In contrast, EMD treats point clouds as probability distributions and determines the minimum cost required to transform one point cloud distribution into another. The spatial distance between points is considered as the cost of moving mass from one point to another. It is defined as follows:

$$EMD_{(X,Y)} = \min_{\emptyset: X \rightarrow Y} \sum_{x \in X} \|\emptyset(x) - x\|_2, \quad (11)$$

where \emptyset represents a bijection between X and Y .

Next, we calculate Minimum Matching Distance (MMD) and Coverage (COV) based on CD and EMD. MMD quantifies the quality of the generated point clouds and assesses the robustness of the model, where a lower MMD value indicates better performance. COV evaluates the diversity of the generated shapes, with a higher COV value being more desirable. In visual assessment, MMD and COV are complementary metrics. Furthermore, to evaluate the naturalness of the samples, we also employ ICTreeI, ICTreeF, and TreeIF, mentioned in Section 3.2, as realism perception metrics.

Implementation Details: During the diffusion phase, we adopted the network parameter settings of SPVD, with a learning rate of 0.001 and 600 epochs. For the virtual tree perception scorers, ICTreeI and ICTreeF, we employed the pre-trained models provided by Romanelis et al. [25]. Furthermore, a GTX 3080Ti GPU was employed to accelerate the neural network training process. The software and hardware configurations of the server are detailed in Table 1.

Table 1. Software and hardware configuration of the server.

Component	Specification
CPU	Intel Core i7-6700 @ 3.40 GHz
GPU	NVIDIA GeForce RTX 3080 TI
RAM	64 GB
Storage	1TB SSD + 4TB HDD
Operating System	Ubuntu 22.04 LTS
NVIDIA Driver	550.135
CUDA Library	11.2
PyTorch	2.0.0 + cu118
Python	3.9

4. Results and Analysis

4.1. Quantitative and Visual Evaluation of Diffusion Results and Analysis

This section presented the results and analysis of the proposed point cloud diffusion method, including loss curves, diffusion process-visualizations, and evaluation metrics such as MMD and COV. These results comprehensively assessed the model's convergence, generation quality, and robustness.

The loss curves for the two datasets, presented in Figure 6, illustrated the training and evaluation loss dynamics over 600 epochs. In both datasets, the loss values decreased significantly during the initial 100 epochs, indicating rapid convergence of the model. Beyond this point, the loss stabilized, fluctuating slightly as the training progressed. The training loss (blue) and evaluation loss (orange) exhibited a consistent trend with minimal divergence, reflecting the model's ability to generalize effectively across both datasets. Notably, TD2 (Figure 6a) started with higher initial loss values compared to TD1 (Figure 6b), indicating a more complex input structure. However, both datasets ultimately achieved comparable loss levels, demonstrating the robustness of the model in handling diverse point cloud structures.

As shown in Figure 7, the qualitative visualizations illustrated the progression of the point cloud diffusion process for tree structure generation. The results depicted a gradual transformation from an initial random noise distribution (leftmost) to fully formed tree structures (rightmost). The intermediate stages depicted the gradual emergence and refinement of branch and trunk structures, with details progressively becoming clearer as the diffusion process advanced. This visualization demonstrated the ability of the diffusion model to generate tree-like structures through an iterative improvement process. Overall, this visualization demonstrated the ability of the diffusion model to generate highly realistic tree-like structures through an iterative improvement process.

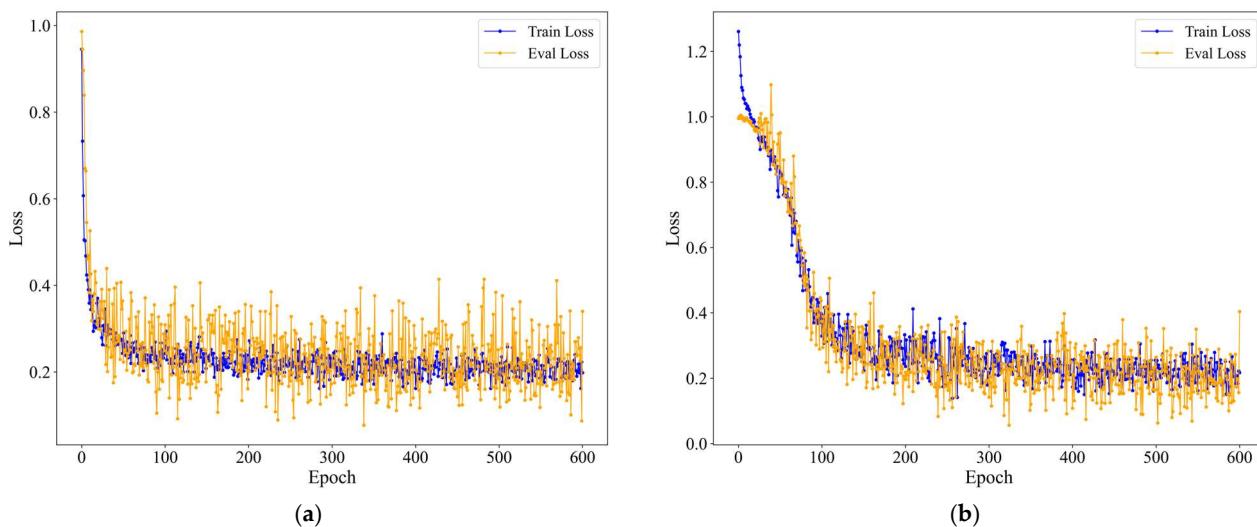


Figure 6. Training and evaluation loss curves for TD1 and TD2. Starting from relatively high initial values, the loss gradually converged over time. **(a)** TD1. **(b)** TD2.

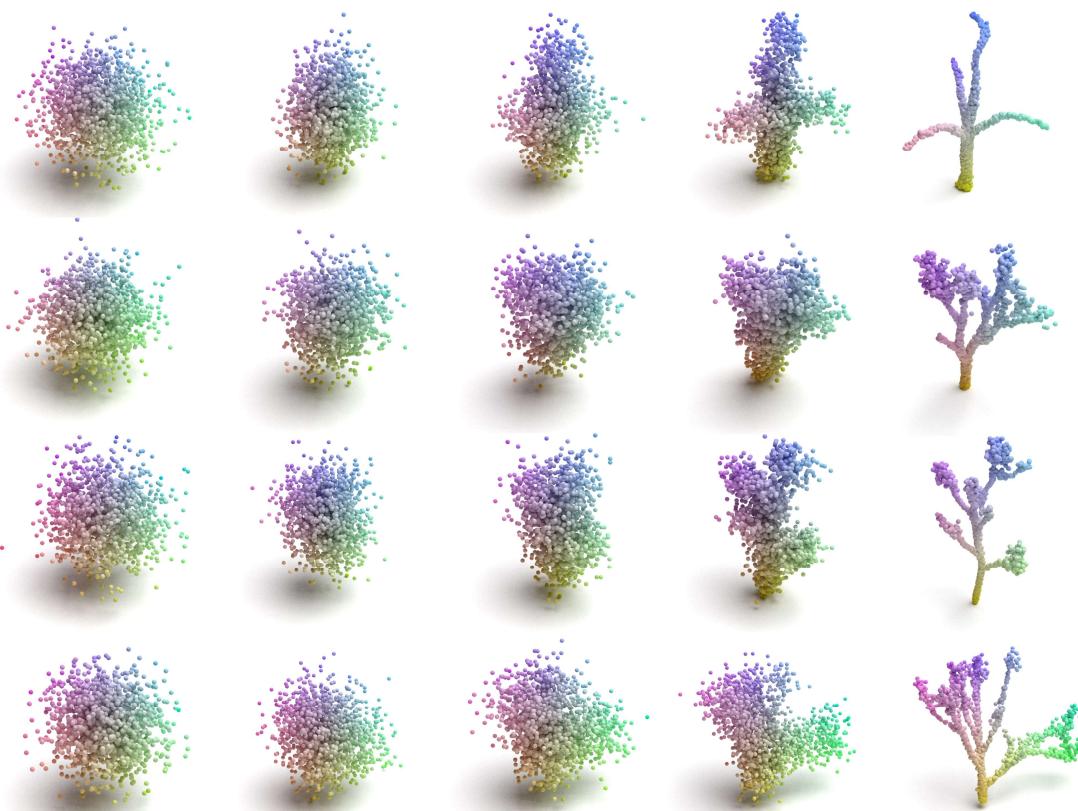


Figure 7. Qualitative visualizations of the diffusion process for tree point cloud generation. The results show the progression from random noise to final tree structures in a left-to-right sequence.

As shown in Table 2, the results revealed that the overall MMD values for TD2 were higher than those for TD1, indicating that the more complex input structures in TD2 posed greater challenges to the model's robustness. Specifically, *Pyrocarpa* exhibited the highest MMD_{CD} and MMD_{EMD} values (0.7445, 0.9815), demonstrating that the model struggled to accurately reproduce its intricate structures. In contrast, simpler tree species in TD1, such as *Apple* and *Walnut*, achieved lower MMD_{CD} and MMD_{EMD} values (0.3261, 0.4319).

and (0.3127, 0.3910), suggesting that the model performed more robustly when generating fewer complex structures.

Table 2. MMD and COV Metrics for each tree species. The two metrics evaluate the robustness of the model and the diversity of the generated samples, respectively.

Type	Tree Species	MMD (↓)		COV % (↑)	
		MMD_{CD}	MMD_{EMD}	COV_{CD}	COV_{EMD}
TD1	<i>Apple</i>	0.3261	0.4319	57.54	65.83
	<i>Cherry</i>	0.4524	0.5748	68.73	71.20
	<i>Ginkgo</i>	0.4167	0.5071	70.56	73.19
	<i>Pine</i>	0.5269	0.7628	73.13	78.64
	<i>Walnut</i>	0.3127	0.3910	58.96	63.44
TD2	<i>Kombolgiensis</i>	0.5917	0.7684	54.42	59.17
	<i>Paliformis</i>	0.4403	0.5435	30.57	36.78
	<i>Pyrocarpa</i>	0.7445	0.9815	62.93	68.31
	<i>Sieberi</i>	0.4551	0.6194	39.28	42.69
	<i>Yilgarnensis</i>	0.5878	0.6125	48.26	55.12

Regarding the COV_{CD} and COV_{EMD} metric, more complex tree species exhibited greater diversity in the generated results. For instance, *Pine* (73.13%, 78.64%) and *Ginkgo* (70.56%, 73.19%) in TD1 achieved the highest COV values, while *Pyrocarpa* (62.93%, 68.31%) in TD2 also demonstrated a high degree of diversity. In contrast, simpler species such as *Paliformis* (30.57%, 36.78%) and *Sieberi* (39.28%, 42.69%) exhibited relatively lower COV values, indicating more limited variability in the generated samples.

Overall, the model exhibited greater robustness, reflected by lower MMD values, when generating simple and smooth tree structures with minimal noise. In contrast, for more complex tree species, the larger structural variability and extended transformation space led to higher sample diversity, as indicated by the increased COV values.

4.2. Comparison of Tree Generation by Different Diffusion Models

Table 3 presents the MMD and COV metrics for different generative methods on the TD1 and TD2 datasets. For both TD1 and TD2, our proposed method consistently achieved the lowest MMD values across both CD and EMD, indicating a better alignment between generated and real samples. Additionally, it outperformed SetVAE and DiT-3D in terms of COV scores, particularly on TD1, where it reached 65.78% for COV_{CD} and 70.46% for COV_{EMD} . This demonstrated that our approach generated more diverse and realistic tree structures compared to SetVAE and DiT-3D methods.

Table 3. MMD and COV of different comparison methods. The two metrics evaluate the robustness of the model and the diversity of the generated samples, respectively.

Method	Tree Dataset	MMD (↓)		COV (↑)	
		MMD_{CD}	MMD_{EMD}	COV_{CD}	COV_{EMD}
TD1	SetVAE	0.6825	0.7754	40.89	55.82
	DiT-3D	0.5826	0.6314	55.74	67.34
	Ours	0.4070	0.5335	65.78	70.46
TD2	SetVAE	0.6918	0.8062	38.71	40.29
	DiT-3D	0.6044	0.7216	46.01	48.16
	Ours	0.5639	0.7051	47.09	52.41

In contrast, SetVAE exhibited the highest MMD values and the lowest COV scores, indicating that its generated samples were less representative of real data. DiT-3D performed better than SetVAE but remained inferior to our method, particularly in MMD_{CD} and COV_{EMD} . The trend remained consistent across both datasets, with TD1 generally showing better performance across all methods compared to TD2, suggesting that TD2 might be a more challenging dataset for generative modeling.

Figure 8 illustrates the tree generation results using different diffusion methods (SetVAE [58], DiT-3D [22], and our proposed approach) on the TD1 and TD2 datasets. Each column corresponded to a generative model, while each row depicted samples from TD1 and TD2. The results indicate that the point clouds generated by SetVAE are relatively dispersed, with an unclear canopy structure. DiT-3D captures the trunk structure but generates uneven density in the fine branches, and at some point, cloud segments are geometrically discontinuous. In contrast, our method produced trees with improved structural coherence, clearer branch connectivity, and a more balanced distribution of fine branches. This trend remained consistent across both datasets, with TD2 trees demonstrating increased branching complexity.

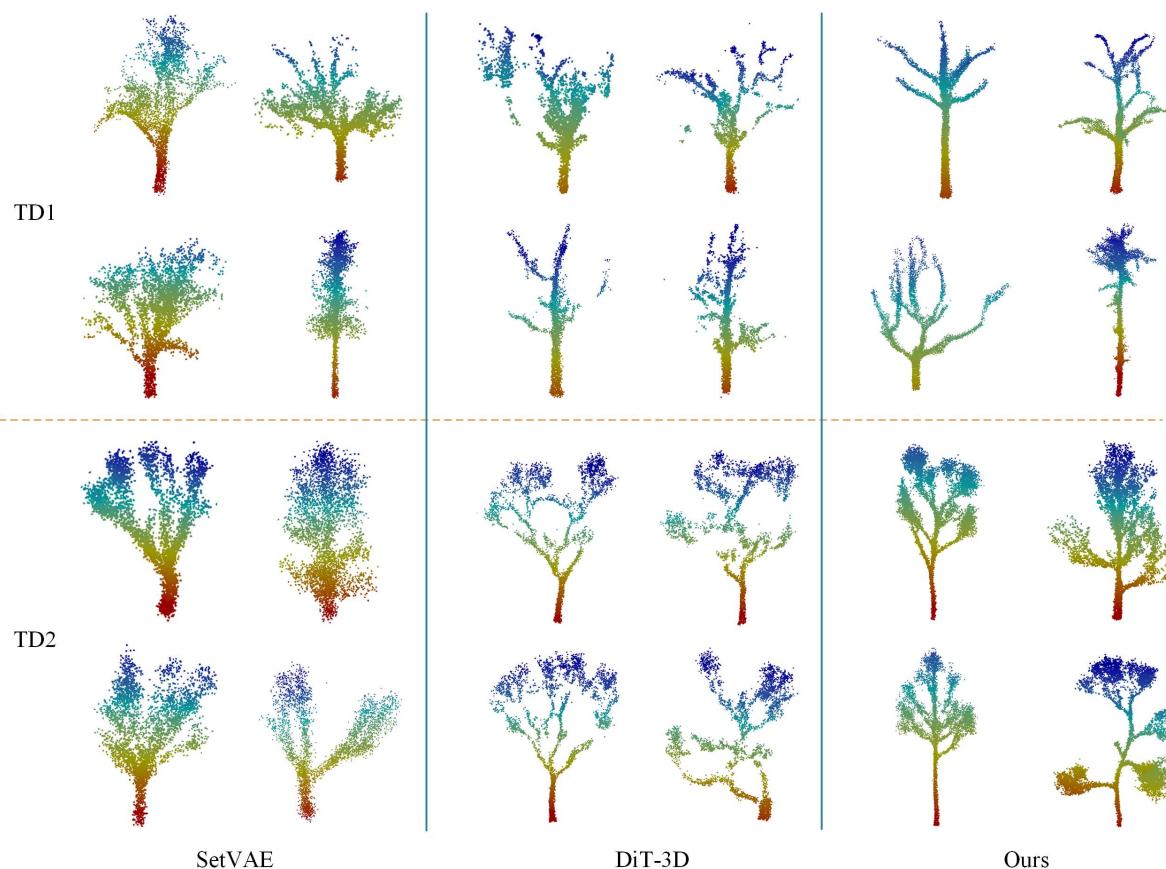


Figure 8. Visualization results of trees generated by different diffusion methods.

Overall, these results highlighted the superior robustness and diversity of our proposed method in tree structure generation, validating its effectiveness in reducing distributional discrepancies and enhancing sample variety.

4.3. TreeIF Filtering and Diversified Generation

Figure 9 and Table 4 present the evaluation results for the ICtreeI, ICtreeF, and TreeIF metrics across two datasets (TD1 and TD2), each comprising five tree species. For TD1, *Ginkgo* exhibited the highest perceptual realism, with ICtreeI, ICtreeF, and TreeIF values

of 0.71, 0.67, and 0.91, respectively, and a threshold T_{TreeIF} of 0.88. In contrast, *Walnut* recorded the lowest ICtreeI (0.43) and ICtreeF (0.42) values, along with a TreeIF value of 0.85 and a threshold T_{TreeIF} of 0.81.

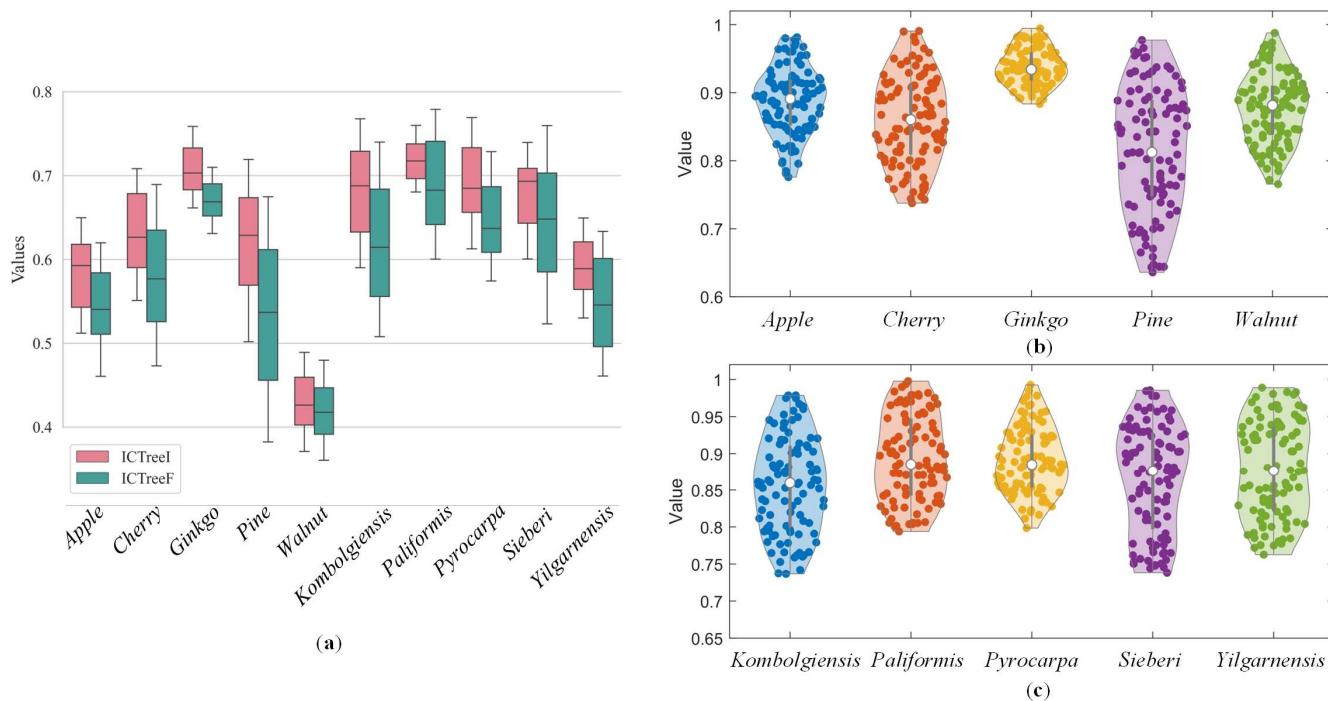


Figure 9. Boxplots of ICtreeI, ICtreeF, and TreeIF. (a) Illustrates the distribution of ICtreeI and ICtreeF values for all tree species in TD1 and TD2. (b) and (c) show the distribution of TreeIF values for tree species in TD1 and TD2, respectively.

Table 4. Realism perception scores (TreeIF) and filtering thresholds for different tree species. The mean and variance were recorded for ICtreeI, ICtreeF, and TreeIF, respectively.

Type	Tree Species	ICtreeI	ICtreeF	TreeIF	T_{TreeIF}
TD1	<i>Apple</i>	0.58 ± 0.02	0.54 ± 0.03	0.86 ± 0.03	0.82
	<i>Cherry</i>	0.63 ± 0.03	0.58 ± 0.04	0.83 ± 0.02	0.78
	<i>Ginkgo</i>	0.71 ± 0.02	0.67 ± 0.02	0.91 ± 0.01	0.88
	<i>Pine</i>	0.62 ± 0.03	0.53 ± 0.05	0.79 ± 0.06	0.72
	<i>Walnut</i>	0.43 ± 0.02	0.42 ± 0.02	0.85 ± 0.03	0.81
TD2	<i>Kombolgiensis</i>	0.68 ± 0.03	0.62 ± 0.05	0.83 ± 0.04	0.76
	<i>Paliformis</i>	0.72 ± 0.02	0.69 ± 0.03	0.87 ± 0.03	0.82
	<i>Pyrocarpa</i>	0.69 ± 0.03	0.64 ± 0.03	0.86 ± 0.03	0.83
	<i>Sieberi</i>	0.68 ± 0.02	0.64 ± 0.04	0.84 ± 0.05	0.77
	<i>Yilgarnensis</i>	0.59 ± 0.02	0.55 ± 0.04	0.85 ± 0.04	0.78

For TD2, *Paliformis* outperformed the other species, achieving ICtreeI, ICtreeF, and TreeIF values of 0.72, 0.69, and 0.87, respectively, with a threshold T_{TreeIF} of 0.82. In contrast, *Yilgarnensis* exhibited the lowest ICtreeI (0.59) and ICtreeF (0.55) values, resulting in a TreeIF value of 0.85 and a threshold T_{TreeIF} of 0.78.

As shown in Figure 9a, the ICtreeI and ICtreeF distributions exhibited moderate variability across species, with *Ginkgo* in TD1 and *Paliformis* in TD2 achieving notably higher scores. Figures 9b and 9c further illustrate the TreeIF value distributions for TD1 and TD2, respectively. Species with higher ICtreeI and ICtreeF values corresponded to elevated TreeIF scores, highlighting the consistency and reliability of the proposed ICtreeF metric as an indicator of realism perception. The threshold T_{TreeIF} also served as the filtering criterion

for generating point clouds for each tree species. Overall, the results demonstrated the consistency and reliability of the proposed ICtreeF metric as an indicator of perceptual realism for the generated tree structures.

Combined with the diffusion model and the TreeIF filtering mechanism, Figure 10 presented the generated point clouds of various tree species in TD1 and TD2, verifying the effectiveness of the proposed method in generating diverse tree shapes. For the simple tree species in TD1 (e.g., *Apple*, *Cherry*, and *Walnut*), the generated point clouds exhibited clear branch structures, and the overall morphology appeared natural and consistent with real trees. For the structurally complex and densely canopied tree species (e.g., *Pine* and *Ginkgo*), the generated trunk structures remained prominent, with only minor noise present in the fine branch regions. A similar pattern was observed for the trees in TD2. Despite the presence of minor noise in the fine branches, this did not affect the overall distinguishability of the generated point clouds or their potential applicability to 3D reconstruction. Subsequent analysis in Section 4.4 will focus on the 3D reconstruction of the generated point clouds and the quantitative evaluation of structural parameters.

Figure 11 presents the generated point cloud samples with TreeIF scores below the threshold. Firstly, the samples with TreeIF scores close to the threshold (e.g., T13, T14, T15, and T16) exhibited relatively complete overall structures but demonstrated localized deficiencies, such as discontinuities in trunks or branches. Among these samples, the TreeIF scores of T13, T14, T15, and T16 were 0.77, 0.72, 0.80, and 0.74, respectively, all slightly below their corresponding thresholds TTTreeIF. Secondly, the samples that exhibited approximate tree-like shapes but structural deformations (e.g., T1, T6, T7, T8, and T11) showed noticeable deformations in bifurcation structures or branch angles, with TreeIF scores close to TTTreeIF. Notably, the TreeIF scores of T6 and T11 were 0.75 and 0.72, respectively, compared to their thresholds of 0.78 and 0.76. Finally, the samples with the lowest TreeIF scores (e.g., T3, T4, T5, T9, T10, and T12) failed to exhibit recognizable tree-like structures. Their point clouds appeared disorganized or fragmented, significantly deviating from the basic tree morphology. In summary, the TreeIF scores effectively reflected the structural integrity of the generated point clouds. Samples with scores close to the threshold closely resembled real trees, whereas those significantly below the threshold exhibited pronounced structural defects or chaotic forms, further validating the reliability and effectiveness of the filtering mechanism.

To validate the stability of Diff-Tree, we conducted 20 independent runs on both the TD1 and TD2 datasets, recording the average values and variance of various evaluation metrics (Table 5). The results showed that, on the TD1 dataset, the point clouds generated by Diff-Tree demonstrated high structural consistency, with MMD_{CD} and MMD_{EMD} values of 0.3927 and 0.5239, respectively, and relatively low variances (± 0.05 and ± 0.04). Additionally, COV_{CD} and COV_{EMD} were 64.26% and 71.69%, indicating high sample diversity. The Pass Rate (PR) was 86.19%, indicating that the majority of generated samples met the quality standards. In contrast, on the TD2 dataset, although the MMD_{CD} and MMD_{EMD} were 0.5438 and 0.6914, indicating that the generated point clouds had relatively more complex structures, the variances remained low (± 0.07 and ± 0.08). The COV_{CD} and COV_{EMD} were 44.38% and 53.55%, with diversity slightly reduced, while the PR was 84.35%, still maintaining a high level. Overall, Diff-Tree demonstrated good stability and robustness across both datasets.

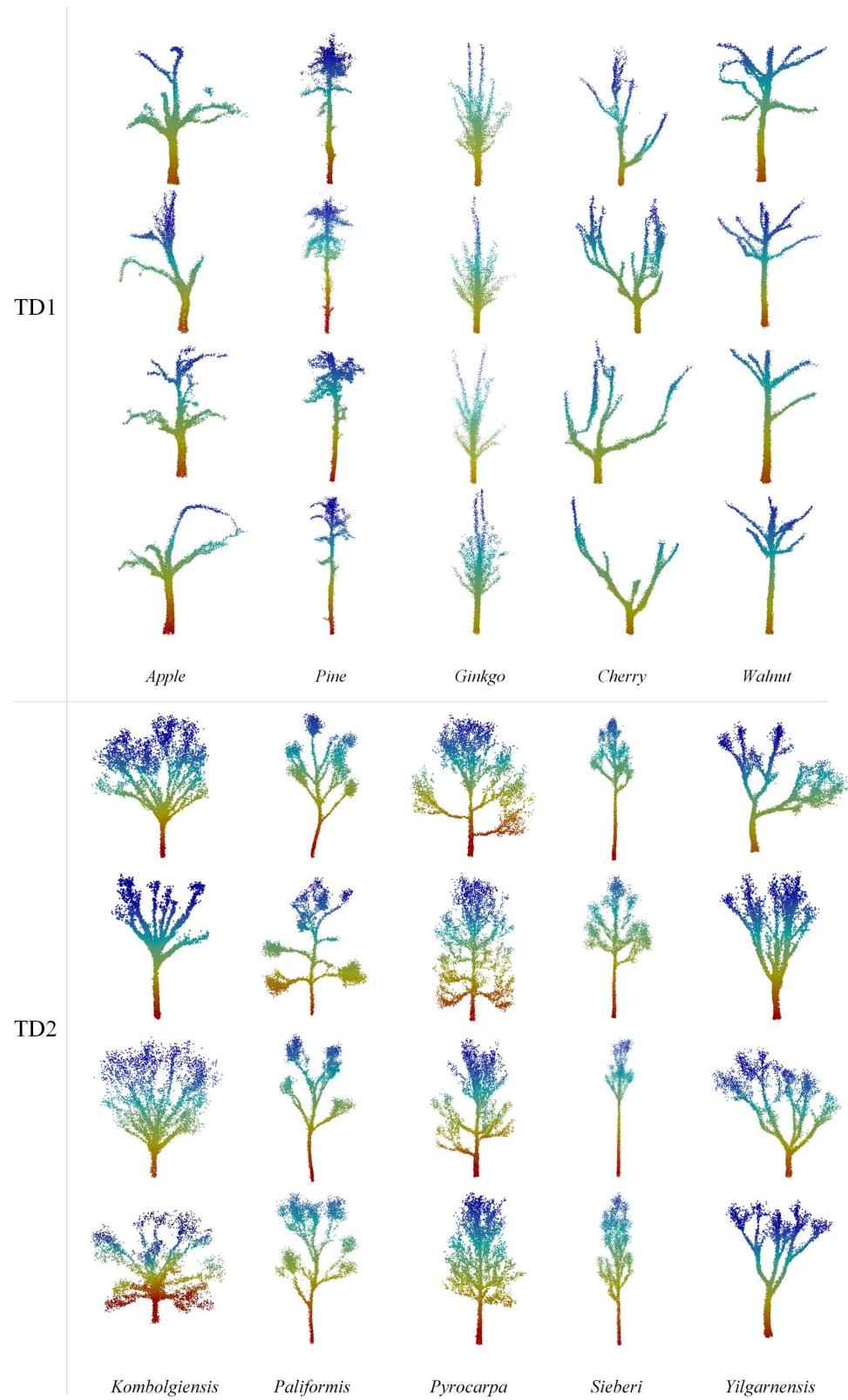


Figure 10. Qualitative visualization of diverse point cloud generation.

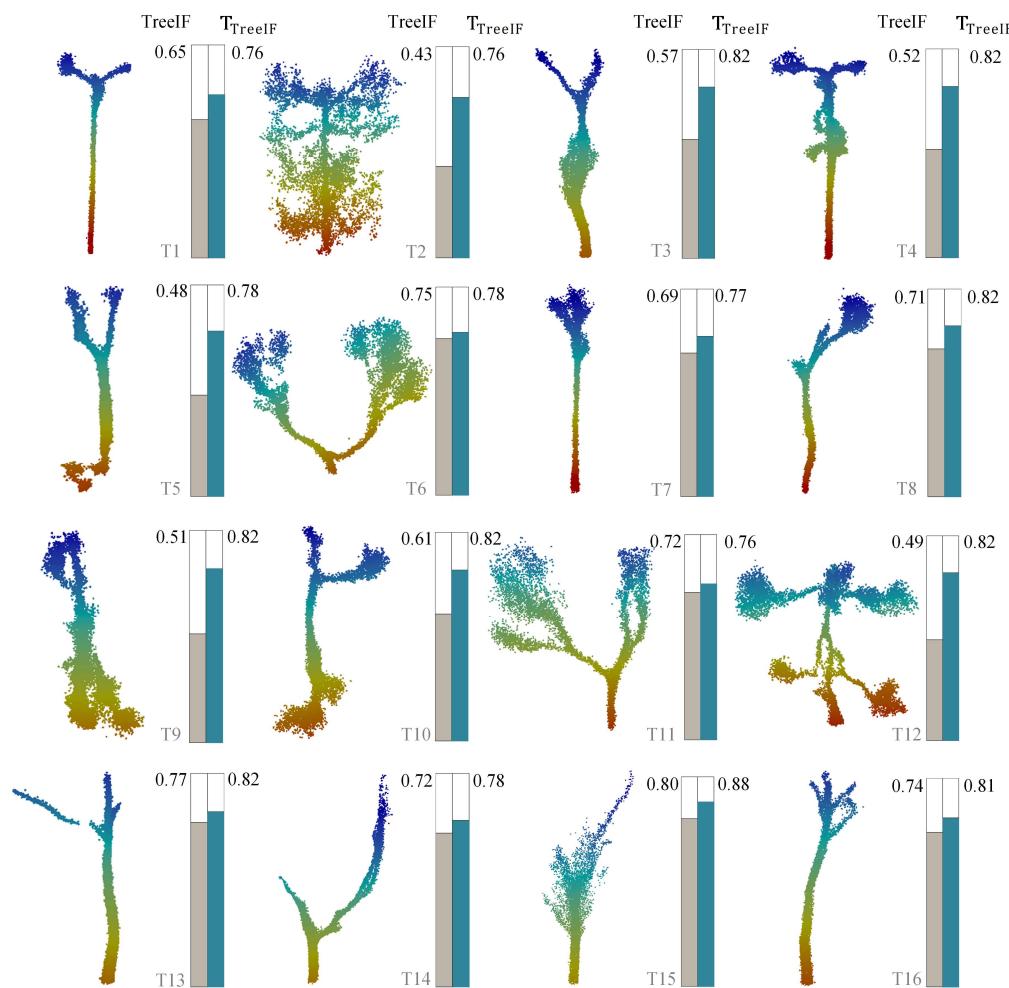


Figure 11. Point clouds below the TreeIF threshold. In each subplot, the gray bar represents the TreeIF value of the point cloud, while the teal bar indicates the TreeIF threshold for the tree species.

Table 5. Average values and variance of evaluation metrics for multiple runs of Diff-Tree. The PR is the pass rate, defined as the number of samples passed using TreeIF filtering divided by the total number of generated samples per batch.

Dataset	MMD_{CD} (\downarrow)	MMD_{EMD} (\downarrow)	COV_{CD} % (\uparrow)	COV_{EMD} % (\uparrow)	PR % (\uparrow)
TD1	0.3927 ± 0.05	0.5239 ± 0.04	64.26 ± 2.42	71.69 ± 3.29	86.19 ± 2.05
TD2	0.5438 ± 0.07	0.6914 ± 0.08	44.38 ± 5.74	53.55 ± 4.47	84.35 ± 1.97

4.4. 3D Reconstruction and QSM Validation

To validate the effectiveness of the generated point clouds in downstream tasks (such as reconstruction), we employed TreeQSM (version 2.4.1) and AdTree (version 1.1.2) to perform a 3D reconstruction of the generated tree point clouds (Figure 12).

Figure 12a presents the visualization of the reconstruction results using AdTree. After 3D reconstruction by the AdTree algorithm, the input point cloud was successfully transformed into geometric structures consistent with natural tree forms, displaying distinct branching structures and canopy hierarchies. Additionally, Figure 12b listed the reconstruction results of the tree point cloud generated from different categories using AdTree and TreeQSM, clearly demonstrating that these generated point clouds were successfully reconstructed by both algorithms and conformed to the topological features of real trees.

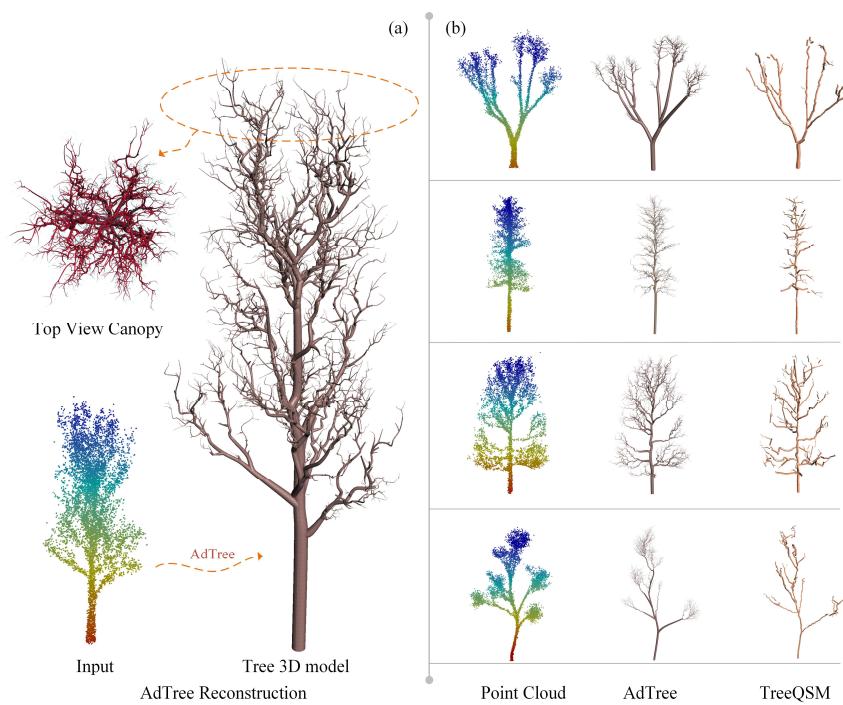


Figure 12. Visualization of point cloud reconstruction. (a) Tree point cloud 3D reconstruction. (b) Results were reconstructed using AdTree and TreeQSM, respectively.

To further validate the structural naturalness and authenticity of the generated tree point cloud, we presented in Figure 13 the structural parameter distributions of the reference samples and generated samples for five tree species (*Kombolgiensis*, *Paliformis*, *Pyrocarpa*, *Sieberi*, and *Yilgarnensis*). The specific parameters included DBH, Number of Branches, Length of Branches, Crown Area, Zenith Angle of Branches, and Diameter of Branches. It can be observed from Figure 13 that:

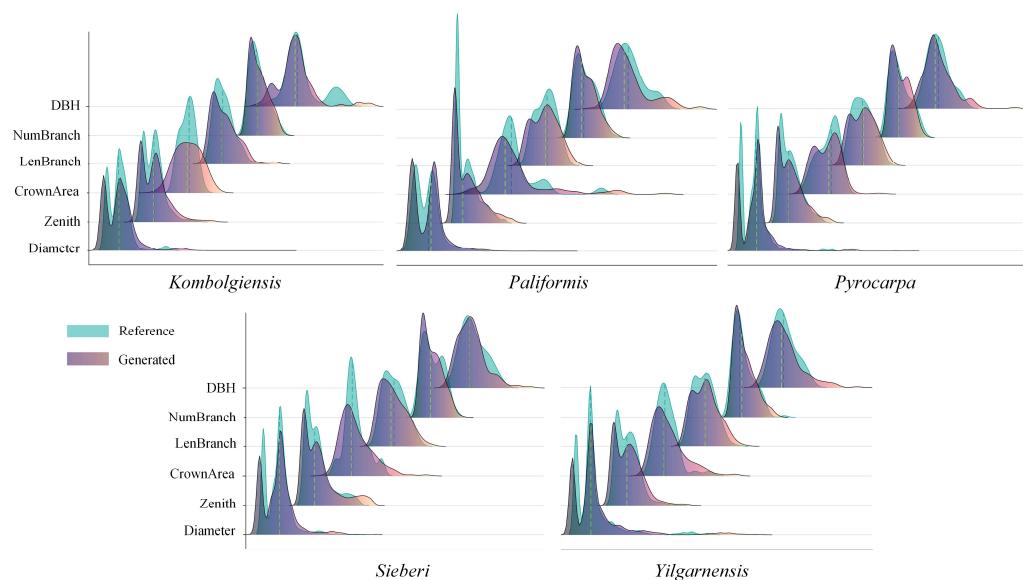


Figure 13. Comparison of tree structural parameters. The figure illustrates six distributions for each tree, corresponding to DBH, Number of Branches (NumBranch), Length of Branches (LenBranch), Crown Area, Zenith Angle of Branches (Zenith), and Diameter of Branches (Diameter). The reference (cyan) represents the distribution of training samples, while the generated (purple) represents the distribution of generated samples. The green dashed line represents the mean.

Overall Consistency: Across all tree species, the parameter distributions of the generated samples aligned closely with those of the reference samples in terms of mean values, distribution patterns (e.g., unimodal or bimodal), and numerical ranges. This indicated that the generated point clouds exhibited geometric structures similar to those of the reference point clouds.

Local Diversity: For some structural parameters, the variance of the generated samples was higher than that of the reference samples, such as length of branches, crown area, zenith angle of branches, and diameter of branches. Specifically, this was reflected in lower peak values and wider distributions. However, DBH and a number of branches remained relatively unchanged. This suggested that while diversifying the generated samples, variations primarily occurred in branch angles and lengths, without any anomalies.

Stability Across Tree Species: The distribution results of the generated samples remained stable across different tree species, further validating the generalization capability of the generation algorithm in handling various tree types.

5. Discussion

5.1. Diff-Tree and TreeIF Performance Analysis

In comparative experiments, Diff-Tree demonstrated a clear advantage, particularly in generating detailed and realistic tree structures. As shown in Figure 8 and Table 3, compared to SetVAE and DiT-3D, SetVAE exhibits limitations in detail recovery, with generated point clouds being relatively dispersed and the canopy structure unclear. DiT-3D, on the other hand, exhibits fragmented point clouds in generating leaves and small branches, failing to capture details effectively, with some areas showing discontinuities in the point cloud. In contrast, Diff-Tree demonstrates superior generation quality in handling complex tree species and detail recovery. This is due to Diff-Tree's use of the sparse point SPVD architecture, which effectively mitigates the effects of non-uniform density, maintaining high-density point clouds in the trunk area while preserving more sparse point clouds in the branches and leaves, thus ensuring structural consistency and clarity of details [23]. Unlike SetVAE and DiT-3D, SetVAE, based on the variational autoencoder method, struggles with non-uniform point cloud density, leading to difficulties in contraction during generation [58]. In contrast, DiT-3D, based on the Transformer architecture, excels in handling geometrically symmetric and uniformly dense point clouds but struggles with complex geometric structures and non-uniformly dense, diversified point clouds [22].

In the sample quality control phase, TreeIF played a crucial role. This mechanism combines the geometric features ICtreeF and visual perception ICtreeI [25] of the tree model to compute TreeIF thresholds based on these characteristics, enabling automatic evaluation of the authenticity of generated samples and filtering point clouds that do not meet the requirements. As shown in Figure 11 and Table 4, TreeIF successfully identifies unrealistic point clouds and removes them. This process significantly improves the quality of the generated point clouds, ensuring that the final output not only aligns with real trees in global structure but also retains excellent detail and depth.

Additionally, Table 5 shows the stability of Diff-Tree, with consistent performance across 20 independent runs on the TD1 and TD2 datasets. The variance fluctuations are minimal, with pass rates of 86.19% and 84.35%, respectively. This indicates that the majority of the generated point clouds meet quality standards, further validating the stability and reliability of Diff-Tree in repeated runs.

5.2. The Generation Capability of Diff-Tree Across Various Tree Species and Structural Diversity

To validate Diff-Tree's generation capability, we used a variety of tree species and datasets from different sources for verification. These datasets include TD1 and TD2 in Section 3.4, as

well as TD3, TD4, and TD5 (with leaves) in Appendix A. Figures 10 and A1–A3 present the generation results of these datasets, while Tables 2 and A1 provide the evaluation results of MMD and COV.

The quantitative results from Tables 2 and A1 indicate that Diff-Tree performs excellently when generating point clouds for tree species with simple structures and low complexity. For example, in the TD1 dataset (in Table 2), *Apple* and *Walnut* show generated point clouds with lower MMD values (*Apple*: 0.3261 for CD, 0.4319 for EMD; *Walnut*: 0.3127 for CD, 0.3910 for EMD) and higher COV values (*Apple*: 57.54% for CD, 65.83% for EMD; *Walnut*: 58.96% for CD, 63.44% for EMD). Similarly, in the TD3 dataset (in Appendix A Table A1), tree species such as *Maple* and *Dracaena* also exhibit generated point clouds with lower MMD values (*Maple*: 0.3329 for CD, 0.4552 for EMD; *Dracaena*: 0.3512 for CD, 0.4219 for EMD) and higher COV values, further demonstrating Diff-Tree's superior performance in generating simple tree species. The same conclusion can also be observed from the generated visualization results in Figures 10, A1 and A2.

However, for more complex tree species, such as *Pine* and *Kombolgiensis* (in Table 2), the quality of the generated models decreases, specifically reflected in higher MMD values and lower COV values. For instance, *Pine* has MMD values of 0.5269 (for CD) and 0.7628 (for EMD), with COV values of 73.13% (for CD) and 78.64% (for EMD), while *Kombolgiensis* has MMD values of 0.5917 (for CD) and 0.7684 (for EMD), with lower COV values (COV for CD: 54.42%, COV for EMD: 59.17%). This suggests that the complexity of branch structures and canopy density increases the generation error, leading to less accurate detail capture compared to simpler tree species.

Additionally, tree species with particularly complex canopy structures, such as those with many branches (TD4) or those with leaves (TD5), face greater challenges in generating point clouds. These results can be observed in Figures A2 and A3, where Diff-Tree successfully generates the trunk and main branches. However, due to the high-density distribution of leaves and the complex geometric structure, the generated point clouds fail to accurately reproduce the distribution of leaves and fine branch details, resulting in discontinuities and incompleteness in the canopy structure. This phenomenon reveals the limitations of Diff-Tree when handling tree species with complex canopy structures and leaves. However, in virtual modeling, branches and leaves are typically modeled separately. For example, methods such as AdTree and TreeQSM primarily model the branches and use leaf insertion algorithms like FaNNI [59] to dynamically add leaves to the model.

In conclusion, Diff-Tree's generation quality decreases as the structural complexity of the tree species increases. This is due to the increased challenges posed by the complexity of branching and denser canopies, particularly when capturing fine details, such as small branch structures and leaf distribution. However, Diff-Tree consistently demonstrates excellent generation capabilities when handling tree point cloud data that include only the trunk, regardless of the species' complexity.

5.3. Analysis of the Usability and QSM Diversification of Generated Samples

To validate the usability of the samples and the diversification of structural parameters, we present the 3D reconstruction results and structural parameter comparisons in Figures 12 and 13 in Section 4.4.

Figure 12 demonstrates that the generated tree point cloud samples can be reconstructed in 3D using AdTree and TreeQSM, capturing both the complete branches and geometric details. There are structural differences between the two reconstruction algorithms, as shown in Figure 12b. This is because AdTree reconstructs trees by initially constructing a minimum spanning graph, then extracting the tree skeleton, and finally employing generalized cylinders to rebuild the tree geometry. In contrast, TreeQSM starts

by clustering the point cloud and subsequently fits cylinders to the clustered point cloud segments. TreeQSM places more emphasis on extracting tree structural parameters, while AdTree focuses on rapidly constructing tree geometries. The differences in their modeling results can be attributed to the distinct logic and intended applications of the two algorithms. However, these differences do not impact the validation of the effectiveness of the point clouds generated by our method.

Additionally, the distribution of structural parameters (DBH, Number of Branches, Length of Branches, Crown Area, Zenith Angle of Branches, and Diameter of Branches) in Figure 13 not only demonstrates the overall structural consistency and local diversity between the generated and reference samples but also illustrates the stability of Diff-Tree in generating specific tree species.

In summary, the point cloud reconstruction results validated that the generated tree point cloud exhibited good geometric interpretability. Additionally, a comparative analysis of tree structural parameters demonstrated that the generated tree point cloud maintained overall structural features consistent with natural patterns while also exhibiting local variations in diversity. This further validated the generation capability and generalization performance of our method.

5.4. Limitations and Outlook

The exploration of the topics addressed in this study is still in its nascent stages. Drawing from our experience, we delineate the present limitations and potential trajectories for future research.

This study primarily investigates stem point clouds devoid of leaves, which presents specific limitations when processing leaf-laden point clouds. This is due to the fact that leaves are typically generated on smaller branches, which are often not captured during LiDAR scanning, leading to geometric discontinuities in the canopy structure of trees with foliage. These complex structural features pose challenges for the model's learning process, resulting in difficulties in achieving accurate representation. Moreover, there is a scarcity of automated evaluation mechanisms for virtual trees. In this study, we incorporated ICTreeI and ICTreeF into a tree realism evaluation system, thus developing a filtering mechanism. However, each tree species necessitates dynamic calculation of the corresponding filtering thresholds. Future research should aim to develop a more adaptive and versatile evaluation mechanism to control the quality of diffusion models more effectively.

A key future direction is the development of 4D growth modeling. By incorporating dynamic temporal growth, where the model evolves over time, intriguing phenomena could emerge in ecological modeling and succession research. Furthermore, introducing additional virtual tree evaluation metrics, such as expert systems, or using more advanced network architectures, will more effectively support data expansion and the development of tree models.

6. Conclusions

In this study, we proposed Diff-Tree, a novel tree point cloud diffuser capable of efficiently generating diverse and high-quality tree point clouds. Specifically, we employed an SPVD-based diffusion architecture to generate a tree point cloud and introduced a filtering mechanism based on the TreeIF. Two sets of tree point clouds with distinct distributions and species categories were used to validate Diff-Tree. Qualitative results demonstrated that Diff-Tree exhibited excellent generative capabilities across various tree species. The diversity of the generated samples and the filtering results further validated the effectiveness of the TreeIF filtering mechanism. Furthermore, we reconstructed the generated point clouds in three dimensions, and through visualization and comparative analysis of tree

structural parameters, we verified the structural diversity and authenticity of the tree point cloud generated by Diff-Tree. This approach not only provided a reliable data foundation for the construction of diverse virtual environments but also showed potential applications in forest digitalization, ecological simulation, and data augmentation for tree-related AI research.

Nevertheless, although our Diff-Tree model successfully generated high-quality and diverse tree point clouds, it should be noted that tree point cloud diffusion is still in its early stages and has certain limitations. First, generating point clouds for a broader range of tree species requires a large and diverse dataset of corresponding point cloud samples, but such high-quality datasets are currently limited. Second, our study focused primarily on leafless tree structures comprising branches and trunks, while the generation of complex point clouds with foliage remains an area for further exploration.

Author Contributions: Conceptualization, H.X., H.L. and Y.H.; Methodology, H.X.; Software, Q.M. and X.Z.; Formal analysis, X.P.; Investigation, X.Z.; Writing—original draft, H.X. and Q.M.; Writing—review and editing, H.L., X.N. and X.P.; Supervision, Y.H.; Funding acquisition, H.L., X.N. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (Grant No. 2023YFC3304000), the Xiong'an New Area Science and Technology Innovation Special Project of the Ministry of Science and Technology of China (2023XAGG0065), and the National Natural Science Foundation of China (Grant No. 42001376).

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

Acknowledgments: We are very grateful to all the reviewers, institutions, and researchers for their help and advice, which helped improve our work.

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

Appendix A

To assess the generalization ability and robustness of Diff-Tree, we utilized point cloud datasets encompassing different tree species and environmental conditions. TD3 [56] and TD4 [60] are point cloud datasets representing only branch structures. Specifically, TD3 consists of 100 samples per species of *Dracaena*, *Maple*, *Spruce*, and *Londo*, all exhibiting significant structural differences. TD4 is a synthetic dataset provided by Lin et al. [60], consisting of 890 synthetically generated tree samples. TD5, incorporating both branch and foliage representations, was released by Dobbs et al. [56] and includes 100 samples per species of *Cherry*, *Maple*, and *Ginkgo*. All datasets were divided into 70% training and 30% testing subsets. The qualitative and quantitative results are summarized in Table A1 and Figures A1–A3.

Table A1. MMD and COV metrics of different datasets.

Type	Tree Species	MMD (↓)		COV (↑)	
		MMD_{CD}	MMD_{EMD}	COV_{CD}	COV_{EMD}
TD3 (branch)	<i>Dracaena</i>	0.3512	0.4219	60.71	67.24
	<i>Maple</i>	0.3329	0.4552	52.64	63.16
	<i>Spruce</i>	0.5024	0.6169	68.19	75.48
	<i>Londo</i>	0.4831	0.5762	64.83	70.24
TD4 (branch)	Synthetic	0.3615	0.4855	58.13	64.25
TD5 (branch + foliage)	<i>Cherry</i>	0.5526	0.6384	40.38	49.36
	<i>Maple</i>	0.4974	0.5851	38.29	42.17
	<i>Ginkgo</i>	0.5733	0.5962	42.81	50.24

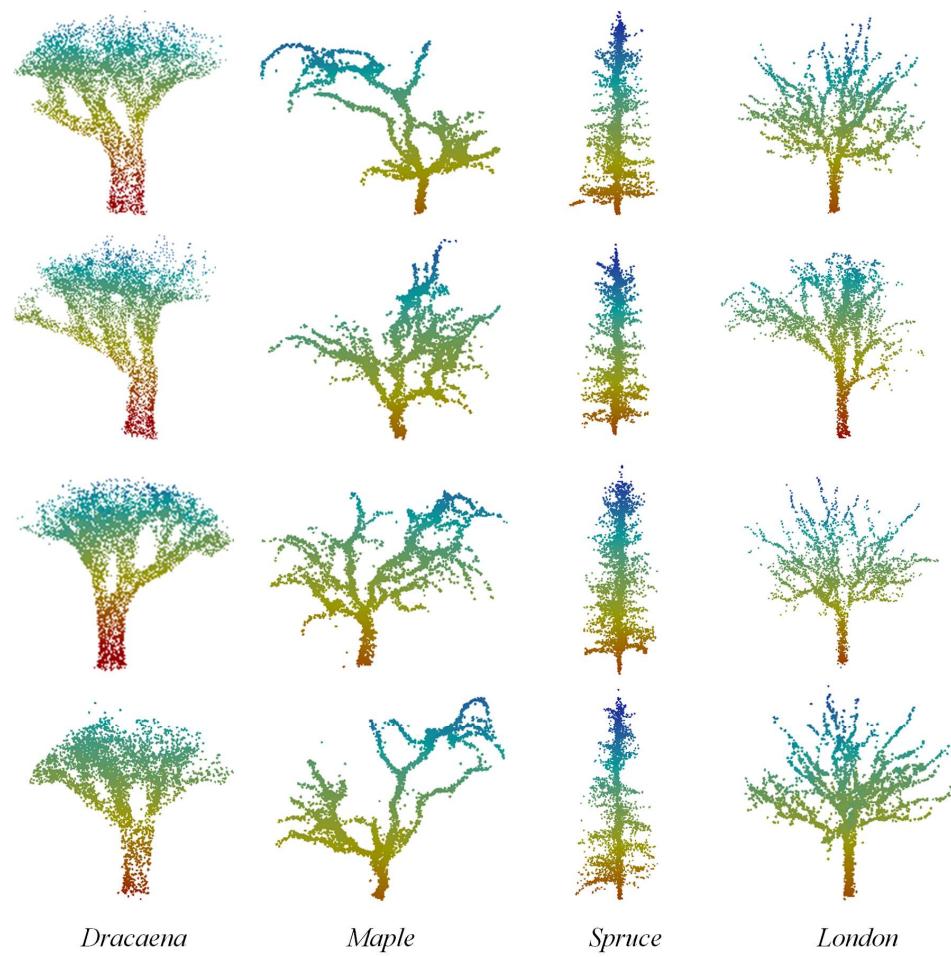


Figure A1. Visualization of trees generated on the TD3 dataset.

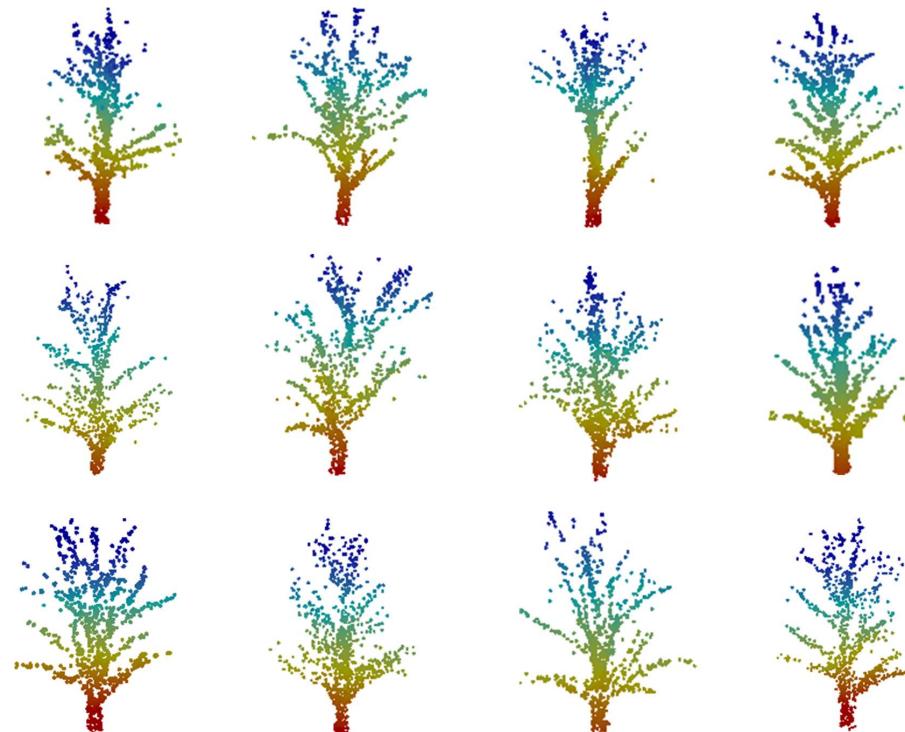


Figure A2. Visualization of trees generated on the TD4 dataset.

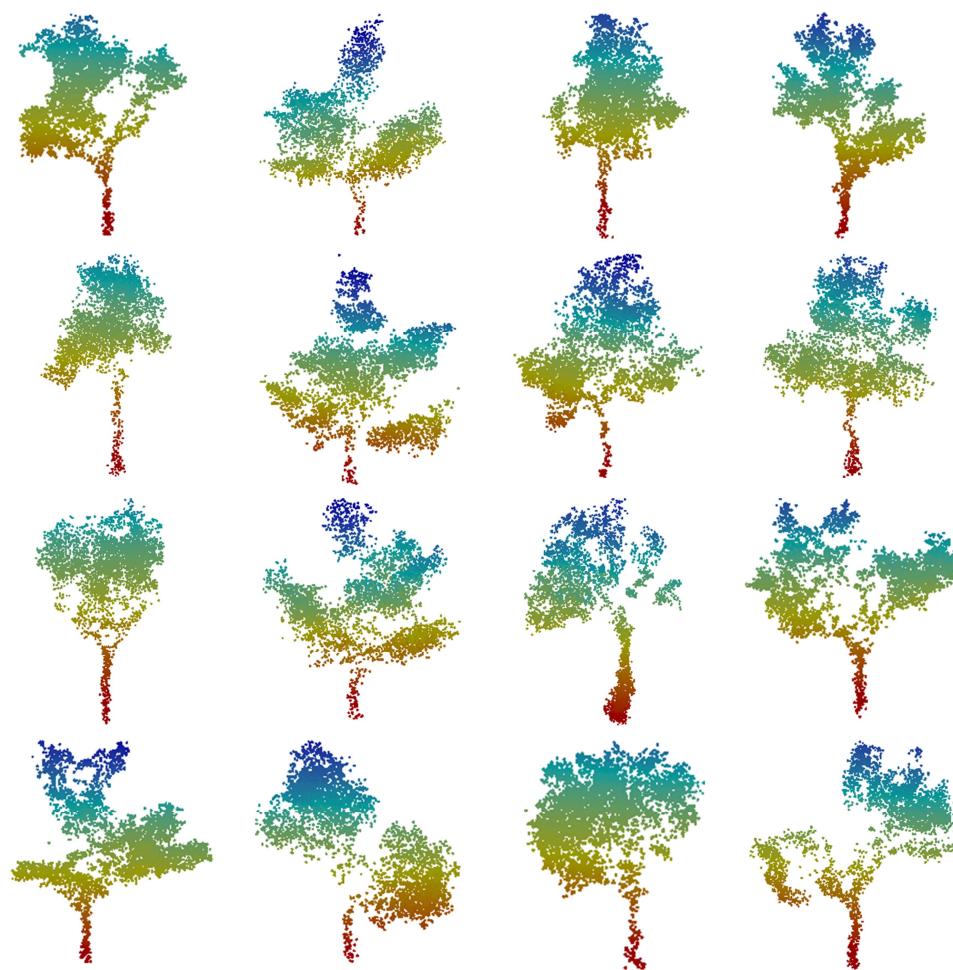


Figure A3. Visualization of trees generated on the TD5 dataset.

References

1. Landsberg, J. Modelling Forest Ecosystems: State of the Art, Challenges, and Future Directions. *Can. J. For. Res.* **2003**, *33*, 385–397. [[CrossRef](#)]
2. Blanco, J.A.; Ameztegui, A.; Rodríguez, F. Modelling Forest Ecosystems: A Crossroad between Scales, Techniques and Applications. *Ecol. Model.* **2020**, *425*, 109030. [[CrossRef](#)]
3. Jia-Tong, W.; Chun-Yue, N.I.U.; Tian-Yu, H.U.; Wen-Kai, L.I.; Ling-Li, L.I.U.; Qing-Hua, G.U.O.; Yan-Jun, S.U. Three-Dimensional Radiative Transfer Modeling of Forest: Recent Progress, Applications, and Future Opportunities. *Chin. J. Plant Ecol.* **2022**, *46*, 1200. [[CrossRef](#)]
4. Li, J.; Yang, B.; Yang, Y.; Zhao, X.; Liao, Y.; Zhu, N.; Dai, W.; Liu, R.; Chen, R.; Dong, Z. Real-Time Automated Forest Field Inventory Using a Compact Low-Cost Helmet-Based Laser Scanning System. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103299. [[CrossRef](#)]
5. Murtiyoso, A.; Holm, S.; Riihimäki, H.; Krucher, A.; Griess, H.; Griess, V.C.; Schweier, J. Virtual Forests: A Review on Emerging Questions in the Use and Application of 3D Data in Forestry. *Int. J. For. Eng.* **2023**, *35*, 29–42. [[CrossRef](#)]
6. Guo, Q.; Su, Y.; Hu, T.; Guan, H.; Jin, S.; Zhang, J.; Zhao, X.; Xu, K.; Wei, D.; Kelly, M.; et al. Lidar Boosts 3D Ecological Observations and Modelings: A Review and Perspective. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 232–257. [[CrossRef](#)]
7. Lei, K.; Zhang, H.; Qiu, H.; Yang, T.; Liu, Y.; Zhang, J.; Hu, X.; Cui, Z. A Novel Strategy for Constructing Large-Scale Forest Scene: Integrating Forest Hierarchical Models and Tree Growth Models to Improve the Efficiency and Stability of Forest Polymorphism Simulation. *Forests* **2023**, *14*, 1595. [[CrossRef](#)]
8. Lin, X.; Li, A.; Bian, J.; Zhang, Z.; Lei, G.; Chen, L.; Qi, J. Reconstruction of a Large-Scale Realistic Three-Dimensional (3-D) Mountain Forest Scene for Radiative Transfer Simulations. *GISci. Remote Sens.* **2023**, *60*, 2261993. [[CrossRef](#)]
9. Okura, F. 3D Modeling and Reconstruction of Plants and Trees: A Cross-Cutting Review across Computer Graphics, Vision, and Plant Phenotyping. *Breed. Sci.* **2022**, *72*, 31–47. [[CrossRef](#)]

10. Li, J.; Wu, W.; Yang, B.; Zou, X.; Yang, Y.; Zhao, X.; Dong, Z. WHU-Helmet: A Helmet-Based Multisensor SLAM Dataset for the Evaluation of Real-Time 3-D Mapping in Large-Scale GNSS-Denied Environments. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5702016. [[CrossRef](#)]
11. Wang, Z. 3D Representation Methods: A Survey. *arXiv* **2024**, arXiv:2410.06475.
12. Qi, J.; Xie, D.; Guo, D.; Yan, G. A Large-Scale Emulation System for Realistic Three-Dimensional (3-D) Forest Simulation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4834–4843. [[CrossRef](#)]
13. Münzinger, M.; Prechtel, N.; Behnisch, M. Mapping the Urban Forest in Detail: From LiDAR Point Clouds to 3D Tree Models. *Urban For. Urban Green.* **2022**, *74*, 127637. [[CrossRef](#)]
14. Fekry, R.; Yao, W.; Cao, L.; Shen, X. Ground-Based/UAV-LiDAR Data Fusion for Quantitative Structure Modeling and Tree Parameter Retrieval in Subtropical Planted Forest. *For. Ecosyst.* **2022**, *9*, 100065. [[CrossRef](#)]
15. Feng, Y.; Shi, X.; Cheng, M.; Xiong, Y. DiffPoint: Single and Multi-View Point Cloud Reconstruction with ViT Based Diffusion Model. *arXiv* **2024**, arXiv:2402.11241.
16. Du, S.; Lindenbergh, R.; Ledoux, H.; Stoter, J.; Nan, L. AdTree: Accurate, Detailed, and Automatic Modelling of Laser-Scanned Trees. *Remote Sens.* **2019**, *11*, 2074. [[CrossRef](#)]
17. Uusitalo, J.; Orland, B. Virtual Forest Management: Possibilities and Challenges. *Int. J. For. Eng.* **2001**, *12*, 57–66. [[CrossRef](#)]
18. Zhou, G.; Wang, B.; Zhou, J. Automatic Registration of Tree Point Clouds From Terrestrial LiDAR Scanning for Reconstructing the Ground Scene of Vegetated Surfaces. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1654–1658. [[CrossRef](#)]
19. Dai, W.; Yang, B.; Liang, X.; Dong, Z.; Huang, R.; Wang, Y.; Pyörälä, J.; Kukko, A. Fast Registration of Forest Terrestrial Laser Scans Using Key Points Detected from Crowns and Stems. *Int. J. Digit. Earth* **2020**, *12*, 1585–1603. [[CrossRef](#)]
20. Wang, W.; Li, Y.; Huang, H.; Hong, L.; Du, S.; Xie, L.; Li, X.; Guo, R.; Tang, S. Branching the Limits: Robust 3D Tree Reconstruction from Incomplete Laser Point Clouds. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *125*, 103557. [[CrossRef](#)]
21. Croitoru, F.-A.; Hondru, V.; Ionescu, R.T.; Shah, M. Diffusion Models in Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10850–10869. [[CrossRef](#)] [[PubMed](#)]
22. Mo, S.; Xie, E.; Chu, R.; Yao, L.; Hong, L.; Nießner, M.; Li, Z. DiT-3D: Exploring Plain Diffusion Transformers for 3D Shape Generation. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 67960–67971.
23. Romanelis, I.; Fotis, V.; Kalogerias, A.; Alexakos, C.; Moustakas, K.; Munteanu, A. Efficient and Scalable Point Cloud Generation with Sparse Point-Voxel Diffusion Models. *arXiv* **2024**, arXiv:2408.06145.
24. Ren, Z.; Kim, M.; Liu, F.; Liu, X. TIGER: Time-Varying Denoising Model for 3D Point Cloud Generation with Diffusion Process. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16 June 2024; pp. 9462–9471.
25. Polasek, T.; Hrusa, D.; Benes, B.; Čadík, M. ICTree: Automatic Perceptual Metrics for Tree Models. *ACM Trans. Graph.* **2021**, *40*, 15. [[CrossRef](#)]
26. Rogowitz, B.E.; Rushmeier, H.E. Are Image Quality Metrics Adequate to Evaluate the Quality of Geometric Objects? In Proceedings of the Human Vision and Electronic Imaging VI, San Jose, CA, USA, 8 June 2001; Volume 4299, pp. 340–348.
27. Lavoué, G.; Larabi, M.C.; Váša, L. On the Efficiency of Image Metrics for Evaluating the Visual Quality of 3D Models. *IEEE Trans. Vis. Comput. Graph.* **2016**, *22*, 1987–1999. [[CrossRef](#)]
28. Wang, Z.; Li, D.; Jiang, R. Diffusion Models in 3D Vision: A Survey. *arXiv* **2024**, arXiv:2410.04738.
29. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 6840–6851.
30. Kulikov, V.; Yadin, S.; Kleiner, M.; Michaeli, T. SinDDM: A Single Image Denoising Diffusion Model. *arXiv* **2024**, arXiv:2211.16582.
31. Xia, B.; Zhang, Y.; Wang, S.; Wang, Y.; Wu, X.; Tian, Y.; Yang, W.; Gool, L.V. DiffIR: Efficient Diffusion Model for Image Restoration. *arXiv* **2023**, arXiv:2303.09472.
32. Zhang, B.; Tang, J.; Niessner, M.; Wonka, P. 3DShape2VecSet: A 3D Shape Representation for Neural Fields and Generative Diffusion Models. *arXiv* **2023**, arXiv:2301.11445. [[CrossRef](#)]
33. Patni, S.; Agarwal, A.; Arora, C. ECoDepth: Effective Conditioning of Diffusion Models for Monocular Depth Estimation. *arXiv* **2024**, arXiv:2403.18807, 28285–28295.
34. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv* **2025**, arXiv:2502.02513.
35. Ikeda, N.; Watanabe, S. *Stochastic Differential Equations and Diffusion Processes*; Elsevier: Amsterdam, The Netherlands, 2014; ISBN 978-1-4832-9615-9.
36. Zeng, X.; Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K. LION: Latent Point Diffusion Models for 3D Shape Generation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10021–10039.
37. Wang, H.; Du, X.; Li, J.; Yeh, R.A.; Shakhnarovich, G. Score Jacobian Chaining: Lifting Pretrained 2D Diffusion Models for 3D Generation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12619–12629.

38. Xu, D.; Jiang, Y.; Wang, P.; Fan, Z.; Wang, Y.; Wang, Z. NeuralLift-360: Lifting an In-the-Wild 2D Photo to a 3D Object With 360deg Views. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 4479–4489.
39. Nichol, A.; Jun, H.; Dhariwal, P.; Mishkin, P.; Chen, M. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv* **2022**, arXiv:2212.08751.
40. Chen, R.; Chen, Y.; Jiao, N.; Jia, K. Fantasia3D: Disentangling Geometry and Appearance for High-Quality Text-to-3D Content Creation. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 22246–22256.
41. Wu, Z.; Wang, Y.; Feng, M.; Xie, H.; Mian, A. Sketch and Text Guided Diffusion Model for Colored Point Cloud Generation. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 8929–8939.
42. Cárdenas, J.L.; Ogayar, C.J.; Feito, F.R.; Jurado, J.M. Modeling of the 3D Tree Skeleton Using Real-World Data: A Survey. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 4920–4935. [CrossRef] [PubMed]
43. Wither, J.; Boudon, F.; Cani, M.; Godin, C. *Structure from Silhouettes: A New Paradigm for Fast Sketch-based Design of Trees*; Wiley Online Library: Oxford, UK, 2009; Volume 28, pp. 541–550.
44. Lindenmayer, A. Mathematical Models for Cellular Interactions in Development I. Filaments with One-Sided Inputs. *J. Theor. Biol.* **1968**, *18*, 280–299. [CrossRef]
45. Okabe, M.; Owada, S.; Igarashi, T. *Interactive Design of Botanical Trees Using Freehand Sketches and Example-Based Editing*; Association for Computing Machinery: New York, NY, USA, 2006; p. 18-es.
46. Liu, Z.; Shen, C.; Li, Z.; Weng, T.; Deussen, O.; Cheng, Z.; Wang, D. Interactive Modeling of Trees Using VR Devices. In Proceedings of the 2019 International Conference on Virtual Reality and Visualization (ICVRV), Hong Kong, China, 18–19 November 2019; pp. 69–75.
47. Yuan, Q.; Huai, Y. Immersive Sketch-Based Tree Modeling in Virtual Reality. *Comput. Graph.* **2021**, *94*, 132–143. [CrossRef]
48. Cheng, Z.-L.; Zhang, X.-P.; Chen, B.-Q. Simple Reconstruction of Tree Branches from a Single Range Image. *J. Comput. Sci. Technol.* **2007**, *22*, 846–858. [CrossRef]
49. Liu, Z. Single Image Tree Reconstruction via Adversarial Network. *Graph. Models* **2021**, *13*, 101115. [CrossRef]
50. Guo, J.; Xu, S. Zhang Realistic Procedural Plant Modeling from Multiple View Images. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 1372–1384. [CrossRef]
51. Livny, Y.; Yan, F.; Olson, M.; Chen, B.; Zhang, H.; El-Sana, J. *Automatic Reconstruction of Tree Skeletal Structures from Point Clouds*; Association for Computing Machinery: New York, NY, USA, 2010; pp. 1–8.
52. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. *arXiv* **2022**, arXiv:2010.02502.
53. Walfish, S. A Review of Statistical Outlier Methods. *Pharm. Technol.* **2006**, *30*, 82.
54. Raumonen, P. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. *Remote Sens.* **2013**, *5*, 30. [CrossRef]
55. Calders, K.; Adams, J.; Armston, J.; Bartholomeus, H.; Bauwens, S.; Bentley, L.P.; Chave, J.; Danson, F.M.; Demol, M.; Disney, M. Terrestrial Laser Scanning in Forest Ecology: Expanding the Horizon. *Remote Sens. Environ.* **2020**, *251*, 112102. [CrossRef]
56. Dobbs, H.; Batchelor, O.; Green, R.; Atlas, J. Smart-Tree: Neural Medial Axis Approximation of Point Clouds for 3D Tree Skeletonization. In Proceedings of the Pattern Recognition and Image Analysis, Alicante, Spain, 27–30 June 2023.
57. 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; pp. 728–737.
58. Kim, J.; Yoo, J.; Lee, J.; Hong, S. Setvae: Learning Hierarchical Composition for Generative Modeling of Set-Structured Data. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 15059–15068.
59. Åkerblom, M.; Raumonen, P.; Casella, E.; Disney, M.I.; Danson, F.M.; Gaulton, R.; Schofield, L.A.; Kaasalainen, M. Non-Intersecting Leaf Insertion Algorithm for Tree Structure Models. *Interface Focus.* **2018**, *8*, 20170045. [CrossRef]
60. Lin, Y.; Liu, J.; Zhou, J. A Novel Tree-Structured Point Cloud Dataset for Skeletonization Algorithm Evaluation. *arXiv* **2020**, arXiv:2001.02823.

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.