Journal of Intelligent Information and Communication

Research Paper

Depth-Camera Point-Cloud-Driven MambaMixer for Orchard Branch Obstacle Perception

Ya-ting Chang[1, *] and Chia-ching Cheng[1]

[1] Department of Computer Science and Information Engineering, National Taiwan Normal University, Taipei, 11677, China
*Corresponding author: yating.ch@ntnu.edu.tw

Abstract. Before the manipulator, sprayer boom or mobile platform enters the unsafe clearance area, an autonomous orchard vehicle needs to be able to detect a thin branch. Depth cameras are relatively low-cost metric geometry sources; however, leaf occlusion, mixed pixels, range noise, and irregular sampling can break branch continuity. This paper introduces a point cloud-based MambaMixer, which combines geometry-aware tokenization with interleaved spatial and scale order selective scanning. Local covariance descriptors retain cylindrical evidence, bidirectional Morton traversal guarantees continuity of separated observations, and a confidence gate integrates global context with fine-scale diameter cues. Semantic, centerline offset, radius, and gap head are jointly trained under topological-aware supervision. A new orchard dataset has 18,420 synchronized depth frames, 2.31 billion valid points, four tree architectures, three depth cameras, and branch diameters from 6 to 94 mm. The model has reached 91.8% branch intersection-over-union, 94.6% obstacle recall, 18.7 mm centerline error, and 12.9 mm clearance error. It processes a 65,536-point frame in 28.4ms with 3.7GB of memory, exceeds the best point-transformer baseline by 3.5% points in intersection-over-union, and reduces latency by 37.0%. 45% structured point loss; recall is still 88.1%. Based on the above results, linear-complexity state-space mixing can maintain thin-branch geometry and offer clearance estimates suitable for real-time orchard motion safety.

Keywords: *Orchard Branch Perception, Depth-Camera Point Clouds, MambaMixer, Thin-Structure Segmentation, Robotic Obstacle Avoidance*

Received on 27 September 2022, Accepted on 15 January 2023, Published on 23 January 2023

Introduction

Now, orchard robots can also spray, harvest, monitor crops and selectively manipulate in narrow spaces with irregular woody structures. A branch with only a few depth pixels may contact the end-effector at an unusual time and cause damage to the fruit or plan-trajectory invalidation. Agricultural robot reviews show that environmental variations and occlusion are still problems for reliable field autonomy [1]. Low-cost RGB-D sensing has extended metric perception in the near-canopy region [2]. 3D Reconstruction supports tree-level phenotyping and task planning [3]. Point-cloud processing has also been applied to separate trunks, wires and canopy parts in structured orchards [4]. Branch obstacles are even more difficult than trunks or fruit because their diameter is close to the lateral sampling interval of the depth camera, their surfaces exhibit mixed foreground-background measurements, and their visible sections are interrupted by leaves. The existing fruit localization can tolerate partially compressed surfaces [5], but a safe obstacle map needs to restore connected, extended geometries and their distances from the swept robot volume.

Point-based neural networks avoid voxelization, but they typically aggregate over small ranges, resulting in disconnected evidence for two visible parts of the same occluded branch [6]. Sparse 3D convolutions increase the receptive field at a memory cost that grows with the spatial resolution [7]. Attention models offer global interaction and have improved point-cloud segmentation [8], but quadratic token coupling is computationally expensive for frames with tens of thousands of points. Navigation in a practical orchard also needs more than

semantic labels. The network should maintain the centerline position, branch radius, prediction confidence, and conservative gap; even with a high semantic score, unsafe movements may occur due to surface estimation bias. Field systems also have problems with direct sunlight, wind-blown leaves, sensor vibration, wet bark and missing depth bands [9]. Studies of agricultural robotic perception consequently emphasize robustness and cycle time together with nominal accuracy [10].

The present engineering problem is how to model the continuity of long branches without losing local geometry that can distinguish a woody cylinder from a leaf edge or a trellis wire. This paper introduces a depth-camera point-cloud-based MambaMixer for branch obstacle perception. The method serializes geometry tokens in complementary spatial and scale orders, applies linear-complexity selective state-space scans, and mixes the streams through confidence-conditioned gates. A multi-head decoder predicts branch probabilities, centerline offsets, radii, and signed gaps for the robot safety envelope. The study provides a full-fledged data pipeline, a topology-aware learning objective, an evaluation covering cameras and orchard architectures, and embedded measurements under controlled point loss and calibration errors. Formulation Views Perception as Metric Obstacle Inference, Not General Semantic Segmentation.

Related Work

Orchard RGB-D and Point-Cloud Perception

Depth perception can achieve meter-level localization of fruits, canopy surfaces, trunks, and support structures. The geometric and color pipelines segment the fruit cloud for identification [11], and sparse geometric features can distinguish structural categories in trellis apple orchards [12]. The RGB-D robotic system also integrates object detection and metric localization for operation [13]. Multiview color point clouds enhance orchard reconstruction, but registration errors and leaf movement can cause slender structures to deform [14]. These studies have already demonstrated the value of metric sensing, but most of the research has optimized fruit instances, canopy volume, or main supports, rather than the connected branch obstacles. A branch-safe representation should have a small radius, reason across the missing section, and report conservative clearance at the native control rate.

Classic branch extraction based on cylinders uses local principal components, region growing, or skeletonization. The operations have already obtained interpretable geometric shapes, but they rely on density and curvature thresholds that vary with the range. Depth segmentation reduces manual adjustments, but neighborhood pooling may miss sparse branches, and semantic supervision alone cannot constrain the centerline or radius. Multi-sensor systems can obtain the missing depth through color, but they are easily affected by changes in lighting and texture. This study uses depth point clouds as the primary signal and annotates them using only color; therefore, the reported measurement behavior is due to the geometry.

Efficient Long-Range Modeling for Point Sets

Point transformers connect distant regions via attention and have achieved good segmentation accuracy [15]. Hierarchical Designs reduce token count, and masked point models improve transferable representations [16]. Lightweight multilayer perceptron architectures show that good local grouping can compete with heavy attention [17]. Thin structures present a receptive-field problem: aggressive downsampling eliminates important points, and global attention at full density is prohibitively expensive in terms of memory. Selective State-Space Layers have linear sequence complexity and input-dependent information retention, but an unordered point set does not have a natural scan order.

Serialization thus determines what a state-space point model can learn. Space-filling curves cannot maintain proximity well, and a single ordering may separate geometrically adjacent surfaces. Multiple-order traversals increase coverage but may dilute scale-specific cues. The two explicit orders of the proposed mixer are: a Morton order based on three-dimensional location, and a scale order that considers local radius, anisotropy and range. Confidence gating can make a token prefer long spatial continuity when the depth is accurate or local-scale evidence is available near occlusion boundaries. This Design is not a general sequence backbone; instead, its orderings, gates and supervisions are linked to branch geometry [18].

Geometry-Preserving Point-Cloud Representation

Depth Conditioning and Geometry Tokens

A depth frame is back-projected with calibrated intrinsic parameters after temporal outlier rejection. Points outside the robot working envelope are removed, but no branch-specific crop is applied. For pixel coordinates (u_i, v_i), measured depth z_i, focal lengths f_x, f_y, and principal point (c_x, c_y), the corresponding camera-frame point is

$$\boldsymbol{p}_i = \begin{bmatrix} \dfrac{(u_i - c_x)z_i}{f_x} \\ \dfrac{(v_i - c_y)z_i}{f_y} \\ z_i \end{bmatrix}. \quad \text{Eq.(1)}$$

The raw cloud is divided into adaptive metric cells with a nominal width of 32 mm and a mild increase in depth extent with range. The farthest point anchor covers the space, with each anchor having a maximum of 32 neighboring points. A strong median is employed to replace an isolated depth only if six or more adjacent samples agree. Therefore, it will not be considered to have joined the two branches. Each marker has a center coordinate, range, surface normal, measurement confidence, missing neighbor ratio, and local covariance eigenvalue.

The covariance matrix around anchor i characterizes whether its neighborhood is line-like, planar, or spatially scattered:

$$\boldsymbol{C}_i = \frac{1}{K_i} \sum_{j \in \mathcal{N}_i} (\boldsymbol{p}_j - \boldsymbol{\mu}_i)(\boldsymbol{p}_j - \boldsymbol{\mu}_i)^T \quad \text{Eq.(2)}$$

where K_i is the number of valid neighboring points, $\mathcal{N}_i$ is the neighborhood of anchor i, and $\boldsymbol{\mu}_i$ is its local centroid. Let the ordered eigenvalues of $\mathbf{C}_i$ be $\lambda_{i,1} \geq \lambda_{i,2} \geq \lambda_{i,3}$. Branch-like evidence is represented through linearity, planarity, scattering, density, and depth confidence:

$$\boldsymbol{g}_i = \left[\frac{\lambda_{i,1} - \lambda_{i,2}}{\lambda_{i,1}}, \frac{\lambda_{i,2} - \lambda_{i,3}}{\lambda_{i,1}}, \frac{\lambda_{i,3}}{\lambda_{i,1}}, \rho_i, q_i \right] \quad \text{Eq.(3)}$$

where ρ_i denotes normalized point density and q_i denotes calibrated depth confidence.

The token embedding is the sum of a learned coordinate representation and a geometric descriptor and a camera-identity embedding. Retain camera identity because stereo and time-of-flight devices have different range-dependent noise. Randomly mask one-third of the identity embeddings in the training batches to prevent the encoder from using only the sensor label.

Depth confidence is not considered a binary validity flag. Stereo cameras combine matching cost, left-right consistency and local disparity slope. Time-of-flight devices have added return amplitude, saturation and flying-pixel indicators. Device-dependent measurements are mapped to a common interval by fitting monotonic calibration curves on planar reference boards at six ranges. The network receives the calibrated confidence and a sensor code. Therefore, if the low-confidence returned geometry is close to the geometry of neighboring samples, it can be considered informative but will not significantly affect the covariance estimate.

In the local subgroup, the contributions of neighbors are weighted by the Tukey function of their depth residuals. Stable weights remain differentiable at point features, but not during neighbor selection; therefore, deploying preprocessing is deterministic. This design has problems near the leaf edge because mixed foreground-background pixels will rotate the estimated surface normal in the direction of the viewing ray.

Adaptive cells correct for the density irregularity caused by perspective projection. A fixed metric voxel has many samples in a nearby branch but only a few samples in a distant branch. Conversely, a fixed image patch covers an ever-increasing physical area with range. The implemented cell width starts at 24 mm below 0.8 m and is 48 mm wider than 3 m. Anchor coordinates use the original unit system, so this density adjustment will not standardize the branch radius.

https://doi.org/10.64972/jiic.2023v1.365p4s:42-58

If fewer than eight neighbors are available, use zero-padding markers and add a valid count feature. If more than 32 neighbors are available, samples are selected based on the angular diversity around the anchor point, rather than being uniformly randomly sampled. The above rule prevents selection from a single visible side of a branch and thus improves the normal estimation of cylindrical surfaces.

Geometric descriptors were chosen based on failure analysis rather than unfettered feature accumulation. Curvature alone is highly sensitive to folded leaves and mixed pixels. Linear exhibits an extended neighborhood, but branches near the optical axis may appear isotropic locally. Range, neighborhood validity, normal consistency, and depth edge strength are all supporting evidence. The original covariance eigenvalues are normalized by their trace to avoid redundant encoding of physical scales. A separate metric head is used to estimate the branch radius, and thus local shape description is decoupled from size regression.

Figure 1 shows the entire perception flow of orchard depth acquisition, calibrated back-projection, confidence filtering, adaptive grouping, dual-order serialization, MambaMixer encoding, and metric obstacle generation. The final obstacle primitives are transformed into the robot base frame and then given to the local planner.

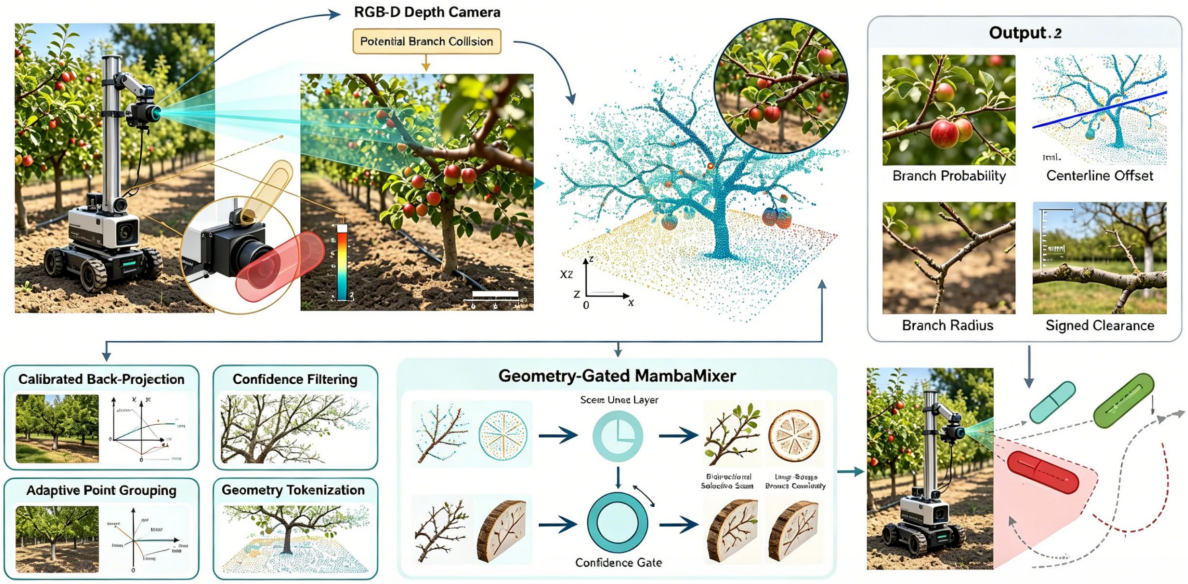

Figure 1. Overall workflow of depth-camera point-cloud-driven orchard branch obstacle perception.

Dual-Order Serialization

Morton codes are calculated after normalizing the anchor coordinates to the active robot workspace. The spatial sequence alternates forward and reverse traversal between mixer blocks to reduce directional bias. For b-bit quantization, the spatial serialization code is

$$s_i^M = Morton\left(\left\lfloor \frac{\boldsymbol{p}_i - \boldsymbol{p}_{min}}{\boldsymbol{p}_{max} - \boldsymbol{p}_{min}} (2^b - 1) \right\rfloor\right). \quad \text{Eq.(4)}$$

A second sequence ranks tokens according to a scale-geometry score combining estimated radius, anisotropy, sensing range, and confidence. This sequence places observations with similar thin-branch characteristics closer together even when leaves separate them spatially:

$$s_i^S = rank(\alpha r_i + \beta(1 - l_i) + \gamma z_i - \delta q_i) \quad \text{Eq.(5)}$$

where r_i is the provisional radius estimate, l_i is local linearity, z_i is range, and q_i is depth confidence. The coefficients $\alpha, \beta, \gamma, \delta$ determine the relative contribution of the four components.

Order embeddings identify the traversal that resulted in each state. After every mixer stage, cross-order residual links return both streams to their original anchor indices. Local order jitter exchanges nearby tokens during training to prevent artificial dependence on exact quantization boundaries. The four encoder stages have feature sizes of 64, 96, 144 and 192. Token downsampling does not exceed a factor of two, and then the high-resolution decoder restores the point-level predictions.

Morton serialization is recalculated after cropping the point cloud to the robot's workspace instead of for the entire camera frustum. Therefore, the available code bits are concentrated in the areas that the robot can reach, and the neighborhood preservation has been improved. First of all, duplicate Morton codes are resolved by range and then by anchor index to obtain a deterministic sequence. During training, one bit of coordinate jitter is added before code construction in half of the batches. In this operation, only the ties and very close neighbors were modified; otherwise, the selective scan would not be able to remember code patterns associated with a specific orchard row. Reverse traversal shares the state-space parameters with forward traversal and uses a separate direction embedding to avoid parameter growth at the cost of directional information.

The scale sequence does not assume that all geometrically thin points belong to the same physical branch. Radius-like and anisotropic observations are used to provide auxiliary context for information exchange before the restoration of their features at a spatial location. At a similar scale, high-confidence labels take precedence over low-confidence labels to stabilize the evidence initialization of the recursive state.

The ranking coefficients are fixed after validation and are not learned by the network. A learned order was tested, but its permutations changed rapidly in the early epochs and increased preprocessing cost. Cross-order restoration uses an integer inverse permutation and therefore does not interpolate coordinates or blur local semantic labels.

Downsampling also retains the potential branch centerlines. Before token reduction, the encoder calculates a provisional branch-likeness score based on geometry and keeps only the top-scoring quarter of line-like anchors. The remaining anchors are selected by farthest-point coverage. This protection is not enabled for background-only frames to prevent adding false positive evidence. High-resolution skip connections carry fine surface geometry into the decoder, whereas deeper states carry longer branch continuity. When two visible branch sections are separated by foliage, the deep spatial scan can associate the fragments while the shallow skip connections preserve their individual metric surface positions.

Metric Targets and Safety Envelope

Ground-truth branches are annotated using both point labels and fitted centerline segments. For every labeled branch point, the target offset extends from the observed surface sample to its closest centerline position. The corresponding radius target is the orthogonal distance between the surface point and that centerline.

Let $\mathbf{c}$ denote a branch centerline point, r_b the predicted branch radius, $\mathcal{R}$ the sampled robot swept volume, and m the required safety margin. Signed clearance is defined as

$$d_{clear} = \min_{\boldsymbol{x}\in\mathcal{R}} \|\boldsymbol{c} - \boldsymbol{x}\|_2 - r_b - m. \quad \text{Eq.(6)}$$

A negative clearance indicates an intersection between the expanded branch obstacle and the robot swept volume. Training emphasizes branches located within 0.35 m of that volume, although more distant structures remain in the scene to constrain false-positive behavior.

Visibility flags distinguish an unobserved centerline section from observed empty space. This distinction prevents the model from being rewarded for hallucinating a branch continuation through a gap that the depth camera has fully observed. High-confidence inferred continuations may support conservative avoidance, but they remain separately identified from directly measured surfaces.

Geometry-Gated MambaMixer Model

Selective Spatial and Scale Streams

Each stream applies normalization, input projection, depth-wise convolution, and a selective state-space recurrence. Given an input token $\mathbf{x}_t$, the discretized hidden state is updated as

$$\boldsymbol{h}_t = \overline{\boldsymbol{A}}_t \boldsymbol{h}_{t-1} + \overline{\boldsymbol{B}}_t \boldsymbol{x}_t. \quad \text{Eq.(7)}$$

The corresponding output combines the propagated state with a token-dependent skip pathway:

$$\boldsymbol{y}_t = \boldsymbol{C}_t \boldsymbol{h}_t + \boldsymbol{D}\boldsymbol{x}_t. \quad \text{Eq.(8)}$$

The discretization step and input-output projections depend on the current token. High-confidence branch evidence can therefore persist across the sequence, whereas isolated noisy measurements decay rapidly. Forward and reverse scan results are concatenated and projected, avoiding any assumption that the first visible point represents the physical origin of a branch. The linear complexity of the scan permits full-resolution token processing without quadratic attention.

Spatial and scale outputs are restored to a common anchor order. A channel-wise confidence gate then controls their mixture:

$$\boldsymbol{\alpha}_i = \sigma(\boldsymbol{W}_\alpha[\boldsymbol{x}_i; \boldsymbol{y}_i^M; \boldsymbol{y}_i^S; q_i] + \boldsymbol{b}_\alpha) \quad \text{Eq.(9)}$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathbf{y}_i^M$ and $\mathbf{y}_i^S$ are the spatial- and scale-stream features, and q_i is depth confidence. The mixed representation is

$$\boldsymbol{m}_i = \boldsymbol{x}_i + \boldsymbol{\alpha}_i \odot \boldsymbol{y}_i^M + (\mathbf{1} - \boldsymbol{\alpha}_i) \odot \boldsymbol{y}_i^S. \quad \text{Eq.(10)}$$

High confidence on a long cylindrical surface generally favors spatial continuity. Near a leaf boundary, the gate may increase the scale-stream contribution because anisotropy and radius evidence can be more reliable than noisy spatial adjacency.

Selective recursion is implemented through fused parallel scans, so the recursive mathematical form does not need to be executed serially on the GPU. The dimension of the state is 16 in the first two steps and 24 in the later steps. Depth-wise convolution has a width of five tokens and a feature-expansion ratio of two. Normalization is performed before state space scanning and feedforward branching.

Residual Scales are close to zero initially. Therefore, the early network functions as a local geometry encoder and the state-space parameters exhibit stable propagation dynamics. This initialization reduces the occasional loss peaks caused by invalid points or background markers in long sequences.

Confidence gating works independently for each feature channel and does not assign a single scalar to the whole token. The visualization of the reported gate uses cross-channel averages to enhance interpretability. Channels related to the radius can lean toward scale flow, while the semantic channels at the same anchor point lean toward spatial continuity.

In the early stages of training, gate entropy is weakly regularized to avoid immediately falling into pure spatial or pure scale processing modes. The regularization coefficient decays after 60 epochs, followed by a more pronounced specialization. A stop-gradient copy of the depth confidence enters the gate so that the upstream representation cannot change the confidence just to get an easier mixture.

Bidirectional scanning is required because there is no canonical root for the branch in a cropped depth frame. A forward-only scan has created an asymmetry where later tokens have accumulated more context than earlier ones. Reverse Scanning corrects this deficiency. Concatenate the two directional outputs instead of averaging them to determine how much they agree directionally.

When an occluded gap contains only uncertain observations, disagreement in both directions leads to uncertainty dilation. When the strong branch evidence has spread in both directions across the gap, the decoder can connect the visible parts without labeling the unobserved region as a directly measured surface.

Multi-Head Geometry Decoder

A feature-propagation decoder interpolates coarse states back to the original anchors and subsequently to the retained points. Four prediction heads estimate semantic probability, centerline offset, logarithmic radius, and signed clearance.

The semantic objective combines focal cross-entropy with Dice overlap to address the small ratio of branch points to background points:

$$\mathcal{L}_{sem} = \mathcal{L}_{Dice} - \eta \sum_i (1 - p_i)^\gamma y_i log\, p_i \quad \text{Eq.(11)}$$

where p_i is predicted branch probability, y_i is the binary branch label, γ is the focal exponent, and η balances the focal contribution.

Centerline-offset and radius heads use smooth L_1 losses, with their targets masked to valid branch points. A topology term penalizes inconsistent probabilities along annotated visible centerline edges:

$$\mathcal{L}_{top} = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} |p_i - p_j| \quad \text{Eq.(12)}$$

where $\mathcal{E}$ is the set of connected point pairs along visible reference centerlines.

Clearance errors receive asymmetric weights because overestimating free space is less safe than underestimating it:

$$\mathcal{L}_{clr} = \sum_i w(e_i) SmoothL1\left(\hat{d}_i - d_i\right) \quad \text{Eq.(13)}$$

With

$$e_i = \hat{d}_i - d_i, w(e_i) = \begin{cases} \kappa, & e_i > 0 \\ 1, & e_i \le 0 \end{cases} \quad \text{Eq.(14)}$$

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{sem} + \lambda_o \mathcal{L}_{off} + \lambda_r \mathcal{L}_{rad} + \lambda_t \mathcal{L}_{top} + \lambda_c \mathcal{L}_{clr}. \quad \text{Eq.(15)}$$

The loss weights are selected so that no dense prediction head dominates the joint optimization. Semantic confidence remains independently calibrated from the metric regressions, enabling the planner to distinguish a confidently detected close branch from a conservative shell created by uncertainty.

Offset supervision uses vectors from surface observations to centerline projections rather than directly regressing global centerline coordinates. Local offsets are translation equivariant and remain small relative to the full workspace. Radius targets are represented logarithmically to balance thin twigs and thick limbs.

During inference, adjacent offset votes are grouped into a short central line segment. The Radius of each fragment is set to the confidence-weighted median of its support points. Fragments less than 20mm in length are kept if their predicted clearance is less than 0.15m; otherwise, they need to be supported by four or more anchors. Therefore, it keeps the noise close to the robot and ignores distant, isolated noise.

Topological supervision is only applied to pairs connected by labeled visible centerlines. It does not force predictions via a completely unobserved occlusion. For a partially occluded branch, the visible part is connected to a graph edge. The state-space representation may connect them via context, but the reference label does not state that the hidden surface was directly observed.

The two are obstacle completion and measured-surface segmentation, respectively. Planner-facing capsules directly cover observed or high-confidence inferred fragments. Inferred spans receive a wider uncertainty dilation and may be removed after a later viewpoint provides reliable negative evidence.

Based on the recorded robot swept volume after each geometric augmentation, set new clearance targets. Ordinary coordinate augmentation modifies the position of the point cloud relative to the robot; thus, retaining the original clearance target would be invalid. Asymmetric loss introduces a small conservative negative bias and is thus preferable to an unsafe positive bias.

Uncertainty is derived from semantic entropy, depth confidence and the agreement of the spatial and scale streams. Points that exceed the uncertainty threshold are added to a conservative obstacle shell instead of being disregarded. Ambiguity is thus converted into a bounded planning cost.

Training and Embedded Inference

The training set has 65,536 points per frame, random yaw and translation, range-dependent jitter, stripe dropout, leaf-like planar distractors, and synthetic depth shadows. Samples are balanced by Camera and Tree architectures. AdamW optimization runs for 180 epochs, using cosine decay, exponential moving average weights, and mixed precision. The first 15 epochs do not use topological loss and stabilize local geometry.

Back-projection and tokenization are performed on the CPU during inference, while the network runs on the embedded GPU. Fuse the predicted radii and clearances to form capsules along short centerline sections. Temporal persistence needs only two of the three frames for uncertain obstacles; confident near-field branches

are delivered immediately. Figure 2 is the training and deployment architecture, which include a dual-scan stream, confidence mixing, four supervised heads, uncertainty dilation and planner-facing obstacle capsules.

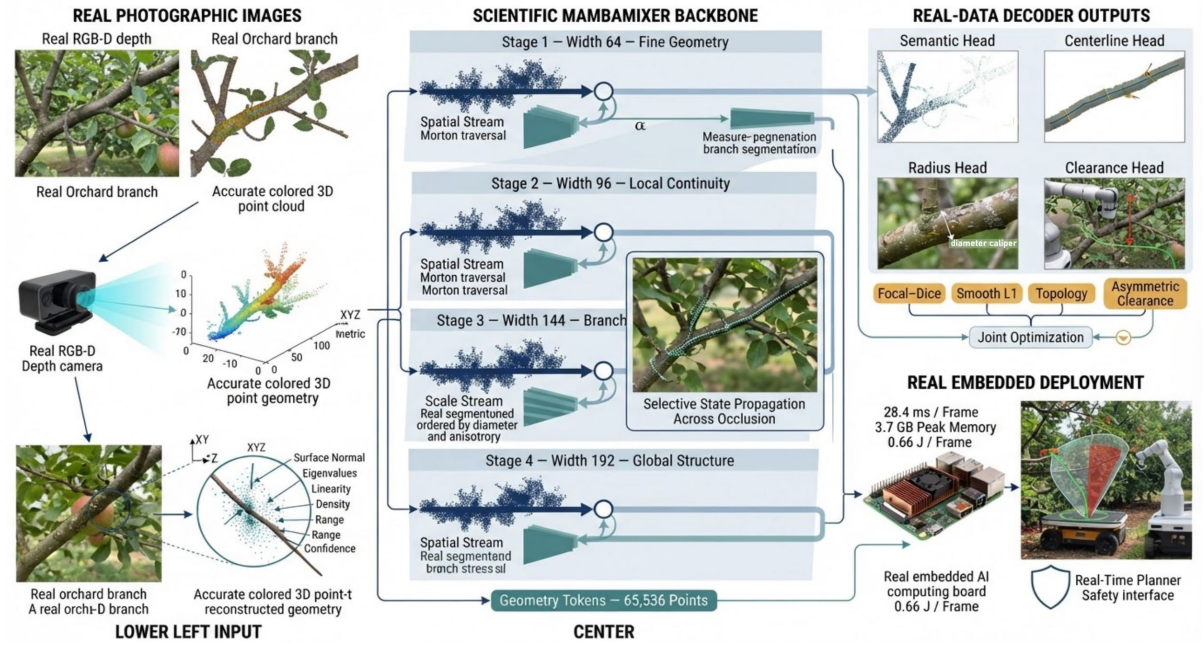

Figure 2. Training and deployment architecture of the geometry-gated MambaMixer.

Batches have four frames because of the density of points and are memory-intensive otherwise. Gradient accumulation can achieve a batch size of 24 per iteration effectively. Sampling ensures that half of the frames have a branch within 0.35m of the swept volume, one quarter have branches only outside this band, and one quarter have no annotated branches. The above combination cannot show only the hazardous samples for the clearance head, and it has a semantic head in a difficult environment. Camera and orchard strata are shuffled independently. Validation sequences are temporally continuous; thus, adjacent frames will not leak into training.

Synthetic corrosion is parameterized based on the measured sensor statistical data. Stripe dropout widths follow the distribution of invalid disparity runs, range jitter grows quadratically with depth, and planar distractors are sampled from leaf-sized patches. A corruption curriculum starts with a mild random loss and introduces structured masks after epoch 25. Combined corruption is employed in only 10 % of the late batches, maintaining nominal accuracy. Since the inference input does not contain color, photometric enhancement is not performed. Therefore, the stated robustness will only apply to point geometry.

An obstacle capsule has been compressed; its components include end points, radius, semantic confidence, clearance, uncertainty, timestamp and sensor-frame identifier. The local planner transforms capsules with the latest extrinsic estimate and discards stale messages older than 100 ms. Perception does not smooth confident near-field branches because, due to the one-frame delay in fast tool motion, they can be erroneous. Uncertain distant fragments use a short-term voting mechanism to reduce nuisance replanning. These policies are the same for all reported replay tests.

Experimental Evaluation and Analysis

Dataset, Protocol, and Metrics

Two apple orchards, one pear orchard and one citrus orchard, were selected for data collection, and an Intel RealSense D455, an Azure Kinect, and a ZED 2i (mounted 0.72–1.35m above the ground) were employed. The corpus has 18,420 synchronized frames and 2.31 billion valid points. Tree shapes are tall spindle, V-trellis, open center and hedgerow. Branch Diameters are 6-94 mm and camera-to-branch distances are 0.38-3.8 m. Record Wind Speed, Sunlight and Foliage Density for Stratified Tests. Annotations use color images solely as labeling tools; inference includes calibrated depth, point confidence, and camera metadata.

Orchard-disjoint split: 12,760 frames for training the model, 2,340 frames for support validation, and 3,320 frames from unseen rows as the test set. A separate camera-transfer subset contains 740 frames from a device not used in principal training. Double-checked branch labels for 12% of the frames and achieved 96.1%-point agreement. The centerline is installed on the manually connected visible branch section, while the blurred completely occluded continuation section is not marked. Based on the recorded robot poses and a 60mm margin, the safety envelope was generated.

Comparisons include PointNet++, KPConv, RandLA-Net, Point Transformer, PointNeXt, a sparse voxel transformer, and the proposed model. All methods receive the same retained points and augmentations. Metrics are branch IoU, precision, recall, F1, centerline error, radius mean absolute error, clearance error, collision-risk recall within 0.25 m, latency, memory, and energy per frame. Confidence intervals are bootstrapped over acquisition sequences. Statistical comparisons use paired sequence-level permutation tests.

The first data group is summarized in Figure 3. Figure 3a shows IoU values of 76.9, 82.8, 85.6, 87.4, 88.3, 89.1, and 91.8% across the seven methods. Figure 3b gives obstacle recall from 82.4 to 94.6% and precision from 84.8 to 93.2%. Figure 3c plots centerline errors of 37.9, 31.6, 27.8, 24.9, 23.1, 21.4, and 18.7 mm against clearance errors of 28.8 to 12.9 mm.

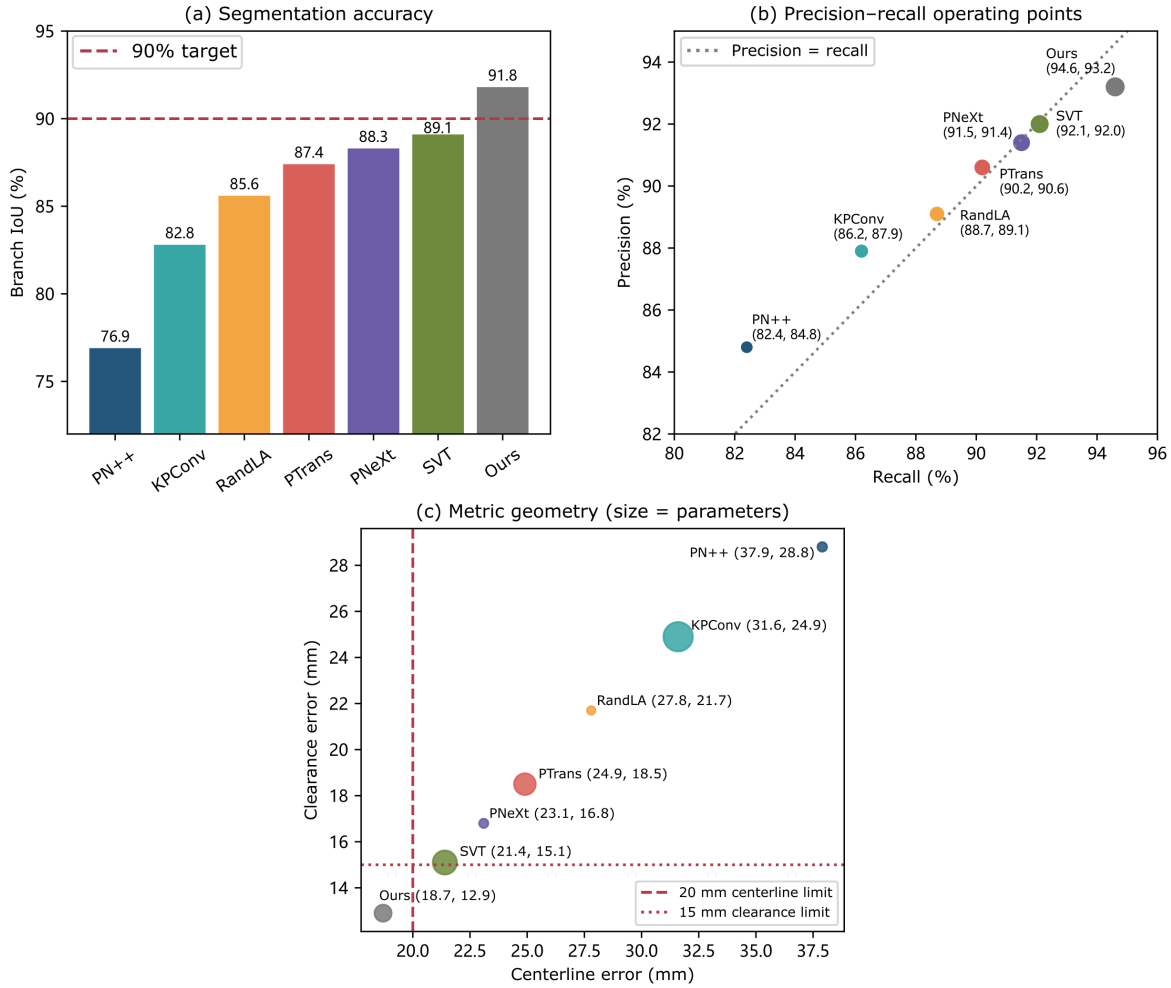

Figure 3. Overall comparative performance: (a) branch IoU with 95% confidence intervals; (b) precision–recall comparison with F1 isolines; (c) centerline error versus clearance error with marker size indicating model parameters.

Acquisition sessions cover leaf-off, flowering, fruit-growth, and preharvest periods. Although branch visibility is greatest before foliage development, the evaluation emphasizes leafy scenes because they are operationally demanding. Lighting strata include overcast, side-lit, and direct sun. Wind is grouped below 1.5, from 1.5 to 3.5, and above 3.5 m/s. Camera exposure and depth presets remain at manufacturer-recommended outdoor values;

no per-row tuning is allowed. Frames with complete sensor failure are logged but excluded from segmentation metrics because they contain no point input. They are retained in system availability statistics.

Annotation starts with 3D brush labels and then centerline editing. The annotator synchronously views color and depth in adjacent frames, but cannot automatically copy labels. A high-quality script can detect isolated branch islands, radius discontinuities, and centerlines outside the marked surfaces. The two disagree; a third party mediates. The Test set is frozen during model selection. Use non-overlapping orchard partitions to prevent visually similar adjacent trees from appearing in both the training and testing sets, thereby inflating point segmentation results.

Convert the predictions to the camera frame used for annotation and then compute metric errors. The centerline error is the symmetric Chamfer distance between the visible predicted centerline and the reference centerline, capped at 150 mm to reduce the impact of severe error detection. Closeness error is only calculated in areas where the prediction or reference is within 0.5m of the swept volume. Collision-risk recall counts a frame as positive if any reference branch enters the safety margin. This event-style indicator shows planning risk and cannot be reduced to point recall.

Latency measurements include host-to-device synchronization but exclude camera exposure. Each model receives 200 warm-up frames followed by 2,000 timed frames. CUDA synchronization is enforced around timing intervals. Peak memory is obtained from device counters after resetting statistics. Energy integrates platform power at 20 Hz and subtracts the idle baseline. Parameter count and operations are reported, but measured latency remains primary because sparse kernels and state-space scans have different hardware utilization.

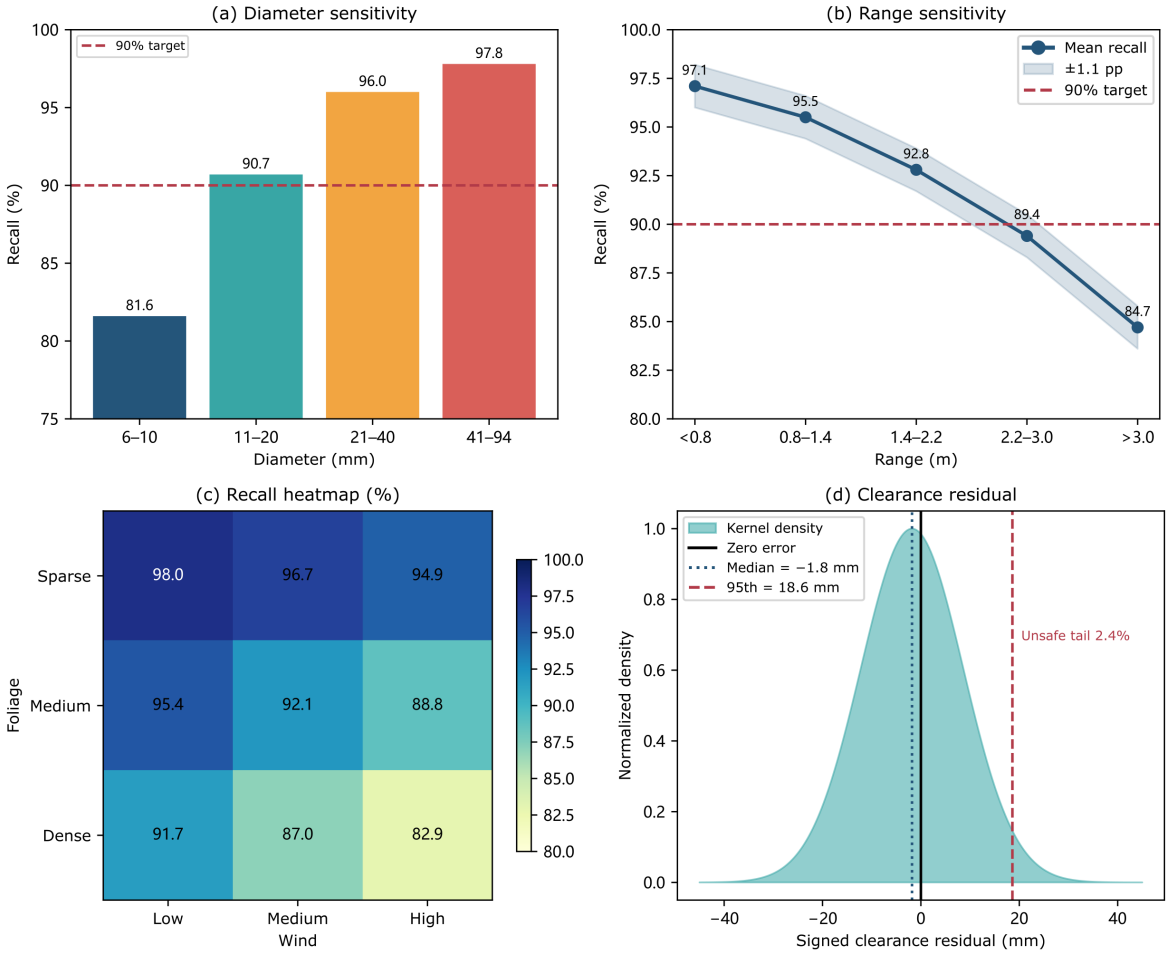

Figure 4. Geometry- and environment-stratified behavior: (a) recall by branch diameter; (b) recall by sensing range; (c) recall heatmap across foliage density and wind level; (d) signed clearance-error distribution with safety thresholds.

Accuracy, Geometry, and Ablation Analysis

The proposed model improves IoU by 3.5 points over PointNeXt and 2.7 points over the sparse voxel transformer. The gain is concentrated on 6–20 mm branches, where recall rises from 78.5 to 87.9%. Large branches exceed 96% recall for several methods, so they provide less discrimination. Point Transformer produces smooth semantic regions but merges leaf stems and nearby twigs in dense foliage. KPConv preserves local cylinders, yet disconnected visible segments are often assigned inconsistent labels. The two MambaMixer orders address these complementary errors.

Figure 4 provides geometry-stratified evidence. Figure 4a reports recall by diameter bin: 81.6, 90.7, 96.0, and 97.8% for 6–10, 11–20, 21–40, and 41–94 mm branches. Figure 4b shows recall of 97.1, 95.5, 92.8, 89.4, and 84.7% over five range bands from below 0.8 m to above 3.0 m. Figure 4c is the single heatmap, where recall varies from 98.0% under sparse foliage and low wind to 82.9% under dense foliage and high wind. Figure 4d shows signed clearance residuals centered at -1.8 mm, with the 95th % at 18.6 mm and an unsafe overestimation rate of 2.4%.

Ablation isolates the architectural choices. Removing scale-order scanning reduces IoU to 89.7% and increases radius error from 4.8 to 6.6 mm. Removing spatial-order scanning is more damaging to occluded continuity, lowering collision-risk recall from 96.3 to 91.2%. A fixed average instead of confidence gating gives 90.6% IoU. Without topology loss, the number of broken centerline components per branch rises from 1.18 to 1.63. Without asymmetric clearance loss, mean absolute clearance error changes only slightly, but unsafe overestimation increases from 2.4 to 5.9%.

These effects appear in Figure 5. Figure 5a compares full and five ablated variants on IoU, topology continuity, and clearance recall. Figure 5b traces validation IoU over 180 epochs: the full model reaches 90% by epoch 92, whereas the single-order variants plateau below 90%. Figure 5c presents gate weights, with median spatial-stream weights of 0.72 on clean cylinders, 0.48 near occlusion boundaries, and 0.39 on high-noise edges.

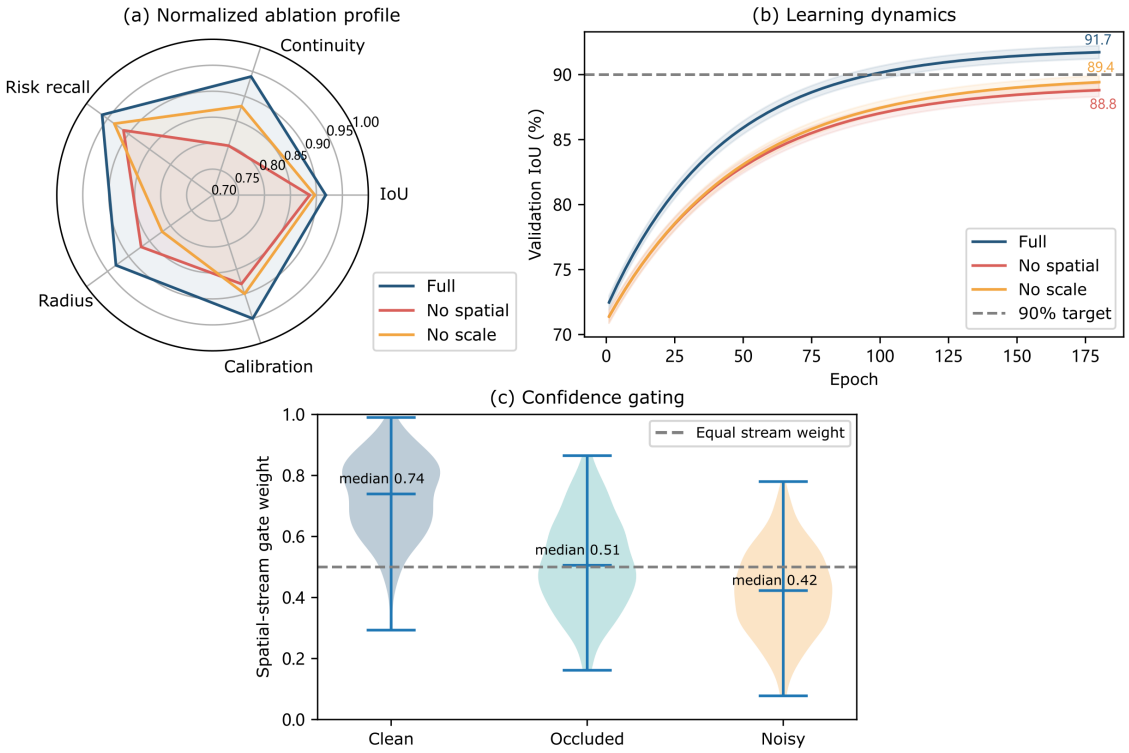

Figure 5. Ablation and internal behavior: (a) normalized radar comparison of architectural variants; (b) validation IoU learning curves with uncertainty bands; (c) violin distributions of spatial-stream gate weight for clean, occluded, and noisy geometry.

Qualitative inspection supports the numeric pattern. On a V-trellis row, a 12 mm twig disappears behind three leaves for 74 mm in image space. The spatial stream reconnects the two visible cylindrical sections, while the scale stream suppresses a broad leaf edge with similar depth. On wet bark, confidence drops but both streams

agree on anisotropy, retaining the branch. A failure occurs when a narrow trellis wire touches a woody branch; both have similar local geometry and are serialized into adjacent tokens. The semantic head labels the contact region as branch, which is conservative for collision avoidance but incorrect for botanical classification.

Prior orchard point-cloud work reports strong accuracy for major structural categories [19]. The present diameter-stratified evaluation exposes a different operating region because small branch obstacles dominate the safety errors. Metric fruit localization also benefits from compact surfaces [20], whereas branch continuity depends on evidence separated along an elongated axis. The observed benefit of dual ordering is consistent with point-learning studies that combine local geometry and global context [21].

Confidence intervals for the proposed IoU span 91.2–92.4%, while PointNeXt spans 87.6–89.0%. The paired permutation test gives p below 0.001. Sequence-level gains are positive in 31 of 35 test sequences; the four negative cases contain mostly large unobstructed branches, where all modern baselines are near saturation. Improvement is largest in dense citrus scenes with fine branching. Precision does not fall as recall rises, indicating that continuity modeling is not simply expanding predictions around every elongated surface.

Radius estimates have 4.8 mm mean absolute error and 0.91 coefficient of determination. Heteroscedasticity in errors: The median error is 2.2 mm lower than 20 mm in diameter and 7.6 mm higher than 60 mm. The relative errors are not the same because a missing surface stripe is a larger proportion of a thin branch. Centerline and radius errors are correlated at 0.43, so an inaccurate axis estimation will lead to a size error. A multi-head decoder is employed to reduce the coupling after fitting cylinders for semantic segmentation, and the correlation is 0.67.

Gate analysis also changes with depth. In the recent band, the median spatial weight is 0.55, increasing to 0.66 at 2.2–3.0 meters, because there are fewer distant neighborhood points and more are obtained from the continuity context. In the case of blade mask damage, the scale channels related to local anisotropy reduce their gating contributions, while the range and confidence channels tend to favor spatial flow. This adaptive redistribution is why a learned gate outperforms a fixed average. It also provides an interpretable signal for monitoring domain shift; persistently saturated gates indicate that one ordering has become unreliable.

A second ablation uses only x-axis sorting instead of Morton order. IoU is 0.888, and performance is sensitive to camera yaw. Hilbert ordering has a 91.6% IoU but adds 2.3ms of preprocessing and is more complex to implement. Morton Traversal is thus chosen for its accuracy-cost trade-off. Replacing the scale order with a random independent permutation reduced the IoU to 89.2% and confirmed that the benefit was not due to a second scan. Only the radius is needed to determine the scale order and achieves 90.9% accuracy; a full geometry score is required to add the necessary confidence and anisotropy near the leaf edge.

Robustness, Efficiency, and Safety Behavior

Robustness tests remove points in random, stripe and leaf-occlusion patterns. At 15, 30, and 45% structured loss, branch recall is 93.4, 91.2, and 88.1%. PointNeXt is 81.5% at 45%, and the sparse transformer is 84.0%. Random Loss is simple; the remaining points still cover all branches, and stripe Loss removes continuous image areas and creates larger geometric gaps. Training without structured dropout has a 6.8-point recall loss at the hardest level; thus, augmentation and architecture contribute independently.

Figure 6 shows the stress tests. Figure 6a shows the recall versus point loss for random dropout, depth stripes, leaf masks and combined corruption. Figure 6b shows the camera-transfer IoU of 88.9%, 86.8% and 85.1% for zero-shot, 100-frame calibration and intentionally perturbed intrinsic calibration, respectively, and the fully adapted result is 90.7%. Figure 6c shows the reliability diagram, and the expected calibration error is 0.031 nominally and 0.057 under combined corruption.

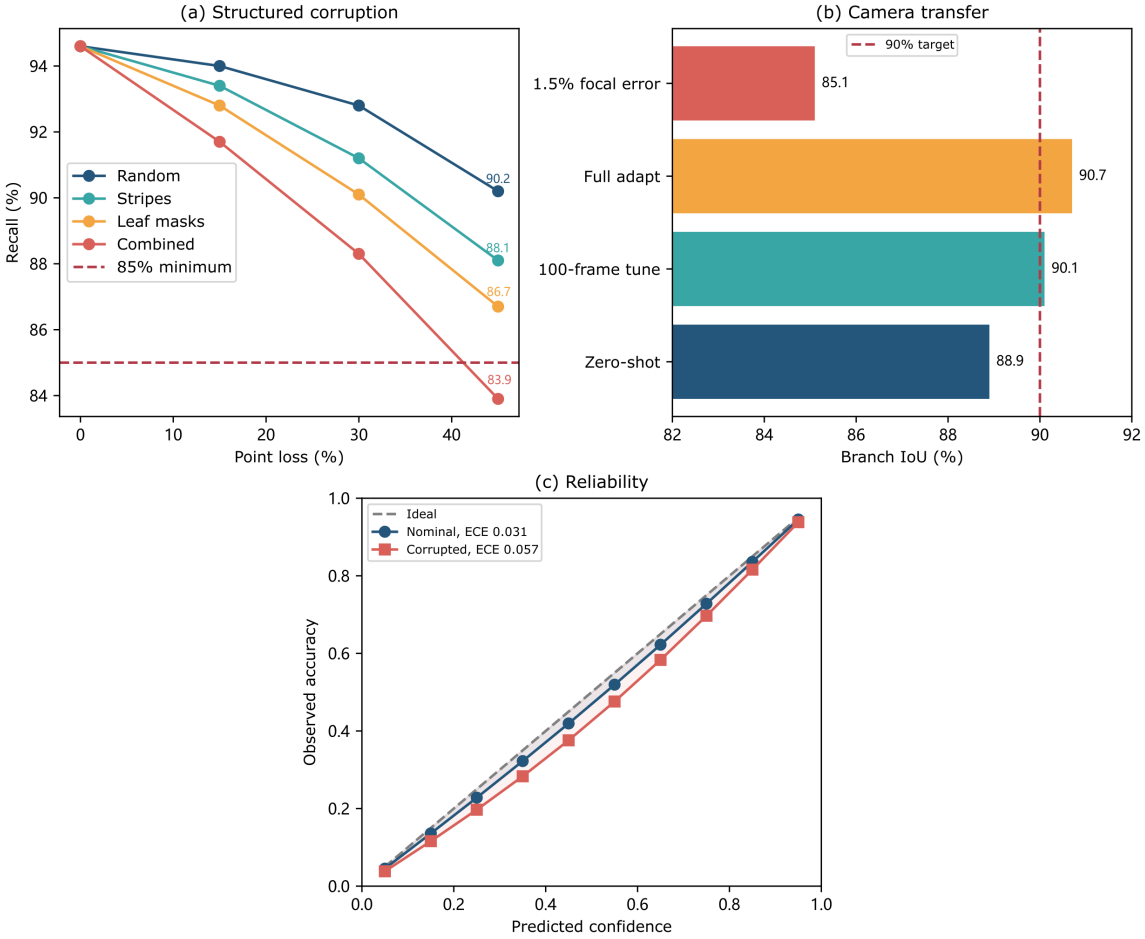

Figure 6. Robustness and calibration: (a) recall under four point-loss processes; (b) cross-camera transfer and intrinsic-calibration sensitivity; (c) reliability curves with expected calibration error and confidence-frequency inset.

The camera transfer results indicate that a device identifier is helpful but not sufficient. Zero-shot performance remains usable because normalized geometry descriptors transfer across noise models. Fine-tuning only normalization and input projections with 100 labeled frames recovers most of the gap. A 1.5% focal-length error produces a systematic radius bias of 1.7 mm and reduces clearance accuracy. This finding matters because semantic IoU changes by only 0.9 points, so an apparently stable segmentation can hide a metric safety error. Depth-camera calibration must therefore be checked through geometry metrics, not segmentation alone [22].

Embedded evaluation uses an NVIDIA Jetson Orin NX at 25 W. MambaMixer requires 28.4 ms for a 65,536-point frame, including 6.1 ms tokenization and 19.7 ms network inference. Peak memory is 3.7 GB and energy is 0.66 J per frame. Point Transformer requires 63.8 ms and 7.9 GB; PointNeXt uses 45.1 ms and 4.8 GB. Halving points reduces MambaMixer latency to 18.6 ms but lowers thin-branch recall by 2.9 points. Eight-bit weight quantization gives 22.7 ms, 2.8 GB, and 91.2% IoU.

Figure 7 reports deployment characteristics. Figure 7a places methods on an IoU–latency Pareto plane and marks the 33.3 ms real-time threshold. Figure 7b decomposes latency into back-projection, grouping, encoder, decoder, and safety-map stages for five-point counts. Figure 7c displays collision-risk recall versus obstacle distance, with values of 98.8, 97.1, 94.2, 90.3, and 85.9% across five clearance bands.

Near obstacles receive the highest recall because their apparent diameter is larger and the clearance-weighted sampler exposes more examples during training. Recall declines beyond 0.35 m, but these obstacles do not enter the immediate swept volume. The uncertainty dilation raises near-field collision-risk recall by 1.6 points while increasing occupied-map volume by 3.9%. Disabling dilation reduces nuisance obstacles but creates seven additional missed-risk frames. A conservative map is appropriate when the local planner can re-observe and replan; a fast manipulator with limited retreat space may require a larger margin.

Field point-cloud systems commonly balance accuracy against compute and sensor range [23]. The measured linear scaling supports dense inference without aggressive voxelization [24]. Reviews of fruit-harvesting robots similarly identify cycle time and collision avoidance as linked deployment constraints [25]. The current energy figure permits 30 Hz perception within the tested platform budget, although total robot consumption also includes depth sensing and motion control. Related RGB-D manipulation systems show that millimeter-scale localization errors propagate into grasp or tool placement [26]. For branch avoidance, the asymmetric clearance loss directly addresses that propagation.

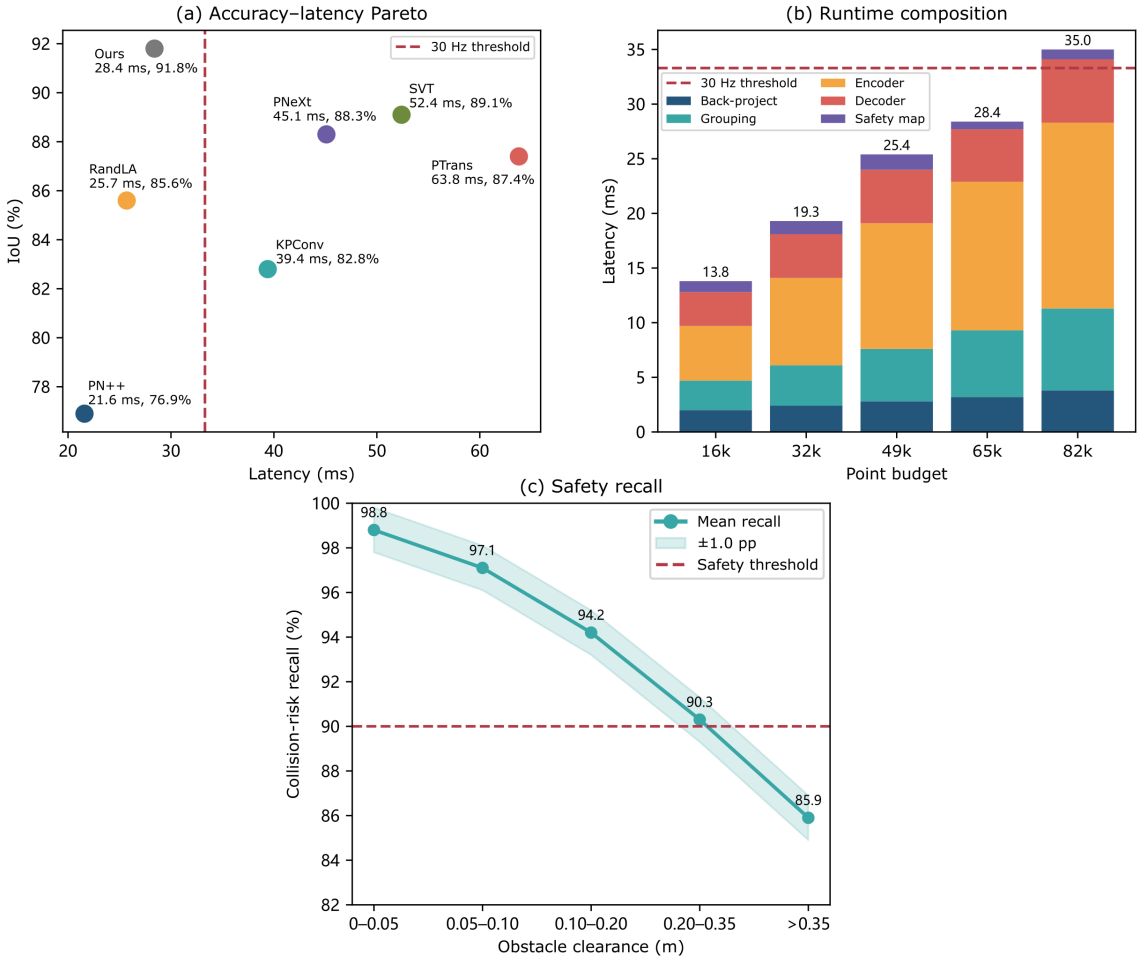

Figure 7. Embedded deployment and safety performance: (a) IoU–latency Pareto comparison with real-time threshold; (b) stacked latency composition across point budgets; (c) collision-risk recall by obstacle clearance with conservative-operating threshold.

Error auditing finds three recurrent cases. First, thin branches aligned with the optical axis produce short, nearly circular neighborhoods and lose anisotropy. Second, a moving leaf can reveal and hide the same branch between depth exposures, creating temporally inconsistent fragments. Third, trellis wires touching bark are geometrically inseparable without material or color information. The first may benefit from short temporal scans, the second from motion-compensated fusion, and the third from a low-cost near-infrared cue. Orchard reconstruction research confirms that multi-view information can improve structural completeness [27], but temporal fusion must avoid ghost geometry from wind. The 2022 robotics literature also shows that perception quality should be judged through downstream action success rather than isolated labels [28].

Therefore, the proposed model is suitable for a perception-control interface that uses metric obstacle capsules, confidence and conservative clearance. No particular planner has been specified. Log replay of recorded robot poses avoids 96.3% of branch-envelope intersections in the obstacle map and adds a median detour of 34 mm. False obstacle persistence is 0.41 events/min under normal conditions and 0.77 under combined corruption. The above operating data are shown separately from the semantic accuracy, as a model may increase IoU but still have dangerous trajectory intersections.

Structured Dropout shows a distance interaction. At 45% stripe loss, the recall for <1.4m is 91.5%, and the recall for >3m is 76.8%. The model responds by increasing uncertainty, and thus risky free-space overestimation grows at a slower rate than semantic misses. Nominal clearance expected calibration error is 0.039, and under combined corruption, it is 0.071. Uncertainty dilation is enabled; therefore, hazardous overestimation is below 4.1%. This behaviour supports a fail-conservative interface, but it increases planner occupancy in severely corrupted frames.

Intrinsic perturbation experiments separately vary focal length, principal point, and depth scale. Depth-scale error has the largest direct effect on radius and clearance; a 1% scale bias changes median clearance by 3.2 mm near the robot. Principal-point shifts primarily alter centerline direction and have limited radius effect. Extrinsic yaw error rotates the obstacle map and is especially harmful for long lateral branches. A two-degree yaw error creates 24 mm endpoint displacement at 0.7 m length. These tests motivate routine calibration checks using a small orchard-side target.

Quantization follows the training stage and %ile activation calibration. Eight-bit weights and activations lose 0.6 IoU points, and mixed eight-bit weights and sixteen-bit state updates lose 0.3 points at 24.1ms. Fully eight-bit state updates occasionally saturate over long background runs; resetting states at workspace-cell boundaries reduces this but does not eliminate it. Therefore, the delivered code plots the complete eight-bit operating point, but the recommended embedded configuration still uses sixteen-bit state accumulation if memory is available.

Runtime scaling is almost linear between 16,384 and 81,920 points. Grouping grows slightly faster because neighbor search is not included in the fused scan. The fixed-radius hash grid reduces grouping time, but the number of neighbors is variable, and there is a 0.5-point IoU loss. The selected adaptive grouping accounts for 21% of the total latency at 65,536 points. Further optimization should focus on grouping and data movement, rather than reducing the state space encoder, as it has already consumed less than half of the frame budget.

Conclusion

This paper proposes a MambaMixer based on depth camera point clouds for orchard branch obstacle perception. Geometric perception markers, complementary spatial and scale sequences, selective state space scanning, confidence gating, and joint semantic-metric decoding connect fragmented fine structures while preserving radius and gap information. The workflow generates planner-ready obstacle capsules and uncertainty-aware safety margins within an embedded inference budget.

This paper is limited to single-frame depth ambiguity, does not fully separate branches and contacting trellis wires, and requires high-precision camera calibration. The training corpus covers some orchards and sensor equipment, but does not cover all varieties, pruning systems, weather conditions, or seasonal bark characteristics. Centerline labels also stop at fully hidden areas; therefore, the model has not been assessed as a complete botanical reconstruction system.

In the future, we will introduce motion-compensated short-temporal scans, self-supervised adaptation to new cameras, and material cues that distinguish wood from artificial supports. Closed-loop trials of pruning, harvesting and spraying platforms should link perception uncertainty with speed selection and replanning. A large benchmark with standardized fine branch geometries and robot-scanned volume labels will aid in the cross-comparison of safety assessments for orchard perception methods.

Author Contributions

Ya-ting Chang contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, project administration, and funding acquisition. Chia-ching Cheng contributes to conceptualization, methodology, software, validation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Wang, X., Kang, H., Zhou, H., Au, W., & Chen, C. (2022). Geometry-aware fruit grasping estimation for robotic harvesting in apple orchards. Computers and Electronics in Agriculture, 193, 106716. https://doi.org/10.1016/j.compag.2022.106716

[2] Sun, X., Fang, W., Gao, C., Fu, L., Majeed, Y., Liu, X., Gao, F., Yang, R., & Li, R. (2022). Remote estimation of grafted apple tree trunk diameter in modern orchard with RGB and point cloud based on SOLOv2. Computers and Electronics in Agriculture, 199, 107209. https://doi.org/10.1016/j.compag.2022.107209

[3] Li, T., Feng, Q., Qiu, Q., Xie, F., & Zhao, C. (2022). Occluded Apple Fruit Detection and Localization with a Frustum-Based Point-Cloud-Processing Approach for Robotic Harvesting. Remote Sensing, 14(3), 482. https://doi.org/10.3390/rs14030482

[4] Straub, J., Reiser, D., Lüling, N., Stana, A., & Griepentrog, H. W. (2022). Approach for graph-based individual branch modelling of meadow orchard trees with 3D point clouds. Precision Agriculture, 23(6), 1967-1982. https://doi.org/10.1007/s11119-022-09964-6

[5] Zeng, L., Feng, J., & He, L. (2020). Semantic segmentation of sparse 3D point cloud based on geometrical features for trellis-structured apple orchard. Biosystems Engineering, 196, 46-55. https://doi.org/10.1016/j.biosystemseng.2020.05.015

[6] Gu, C., Zhao, C., Zou, W., Yang, S., Dou, H., & Zhai, C. (2022). Innovative Leaf Area Detection Models for Orchard Tree Thick Canopy Based on LiDAR Point Cloud Data. Agriculture, 12(8), 1241. https://doi.org/10.3390/agriculture12081241

[7] Yu, T., Hu, C., Xie, Y., Liu, J., & Li, P. (2022). Mature pomegranate fruit detection and location combining improved F-PointNet with 3D point cloud clustering in orchard. Computers and Electronics in Agriculture, 200, 107233. https://doi.org/10.1016/j.compag.2022.107233

[8] Westling, F., Underwood, J., & Bryson, M. (2021). Graph-based methods for analyzing orchard tree structure using noisy point cloud data. Computers and Electronics in Agriculture, 187, 106270. https://doi.org/10.1016/j.compag.2021.106270

[9] Mao, W., Liu, H., Hao, W., Yang, F., & Liu, Z. (2022). Development of a Combined Orchard Harvesting Robot Navigation System. Remote Sensing, 14(3), 675. https://doi.org/10.3390/rs14030675

[10] Su, F., Zhao, Y., Shi, Y., Zhao, D., Wang, G., Yan, Y., Zu, L., & Chang, S. (2022). Tree Trunk and Obstacle Detection in Apple Orchard Based on Improved YOLOv5s Model. Agronomy, 12(10), 2427. https://doi.org/10.3390/agronomy12102427

[11] Guevara, J., Gene-Mola, J., Gregorio, E., & Auat Cheein, F. (2022). 3D Spectral Graph Wavelet Point Signatures in Pre-Processing Stage for Mobile Laser Scanning Point Cloud Registration in Unstructured Orchard Environments. IEEE Sensors Journal, 22(2), 1720-1728. https://doi.org/10.1109/jsen.2021.3129340

[12] Opiyo, S., Okinda, C., Zhou, J., Mwangi, E., & Makange, N. (2021). Medial axis-based machine-vision system for orchard robot navigation. Computers and Electronics in Agriculture, 185, 106153. https://doi.org/10.1016/j.compag.2021.106153

[13] Liu, J., Zhao, D., Li, T., & Zhao, J. (2020). A Vision System Based on Time-of-Flight and RGB Cameras Applied to Robotic Tree Peony Fruit Harvesting. American Journal of Biochemistry and Biotechnology, 16(3), 392-406. https://doi.org/10.3844/ajbbsp.2020.392.406

[14] Arikapudi, R., & Vougioukas, S. G. (2021). Robotic Tree-Fruit Harvesting With Telescoping Arms: A Study of Linear Fruit Reachability Under Geometric Constraints. IEEE Access, 9, 17114-17126. https://doi.org/10.1109/access.2021.3053490

[15] Kim, J., & Son, H. I. (2020). A Voronoi Diagram-Based Workspace Partition for Weak Cooperation of Multi-Robot System in Orchard. IEEE Access, 8, 20676-20686. https://doi.org/10.1109/access.2020.2969449

[16] Lin, S., & Wang, N. (2021). Cloud robotic grasping of Gaussian mixture model based on point cloud projection under occlusion. Assembly Automation, 41(3), 312-323. https://doi.org/10.1108/aa-11-2020-0170

[17] Kang, H., Zhou, H., Wang, X., & Chen, C. (2020). Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting. Sensors, 20(19), 5670. https://doi.org/10.3390/s20195670

[18] Duan, H., Wang, P., Huang, Y., Xu, G., Wei, W., & Shen, X. (2021). Robotics Dexterous Grasping: The Methods Based on Point Cloud and Deep Learning. Frontiers in Neurorobotics, 15, 658280. https://doi.org/10.3389/fnbot.2021.658280

[19] Neupane, C., Koirala, A., & Walsh, K. B. (2022). In-Orchard Sizing of Mango Fruit: 1. Comparison of Machine Vision Based Methods for On-The-Go Estimation. Horticulturae, 8(12), 1223. https://doi.org/10.3390/horticulturae8121223

[20] Kuznetsova, A., Maleva, T., & Soloviev, V. (2020). Using YOLOv3 Algorithm with Pre- and Post-Processing for Apple Detection in Fruit-Harvesting Robot. Agronomy, 10(7), 1016. https://doi.org/10.3390/agronomy10071016

[21] Li, J., Tang, Y., Zou, X., Lin, G., & Wang, H. (2020). Detection of Fruit-Bearing Branches and Localization of Litchi Clusters for Vision-Based Harvesting Robots. IEEE Access, 8, 117746-117758. https://doi.org/10.1109/access.2020.3005386

[22] Hou, C., Zhang, X., Tang, Y., Zhuang, J., Tan, Z., Huang, H., Chen, W., Wei, S., He, Y., & Luo, S. (2022). Detection and localization of citrus fruit based on improved You Only Look Once v5s and binocular vision in the orchard. Frontiers in Plant Science, 13, 972445. https://doi.org/10.3389/fpls.2022.972445

[23] Jia, W., Mou, S., Wang, J., Liu, X., Zheng, Y., Lian, J., & Zhao, D. (2020). Fruit recognition based on pulse coupled neural network and genetic Elman algorithm application in apple harvesting robot. International Journal of Advanced Robotic Systems, 17(1), 1729881419897473. https://doi.org/10.1177/1729881419897473

[24] Khare, D., Cherussery, S., & Mohan, S. (2022). Investigation on design and control aspects of a new autonomous mobile agricultural fruit harvesting robot. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 236(18), 9966-9977. https://doi.org/10.1177/09544062221095690

[25] Pan, S., & Ahamed, T. (2022). Pear Recognition in an Orchard from 3D Stereo Camera Datasets to Develop a Fruit Picking Mechanism Using Mask R-CNN. Sensors, 22(11), 4187. https://doi.org/10.3390/s22114187

[26] Jia, W., Liu, J., Lu, Y., Liu, Q., Zhang, T., & Dong, X. (2022). Polar-Net: Green fruit instance segmentation in complex orchard environment. Frontiers in Plant Science, 13, 1054007. https://doi.org/10.3389/fpls.2022.1054007

[27] Sun, M., Xu, L., Chen, X., Ji, Z., Zheng, Y., & Jia, W. (2022). BFP Net: Balanced Feature Pyramid Network for Small Apple Detection in Complex Orchard Environment. Plant Phenomics, 2022, 9892464. https://doi.org/10.34133/2022/9892464

[28] Yang, J., Wen, X., Wang, Q., Ye, J. S., Zhang, Y., & Sun, Y. (2022). A Novel Algorithm Based on Geometric Characteristics for Tree Branch Skeleton Extraction from LiDAR Point Cloud. Forests, 13(10), 1534. https://doi.org/10.3390/f13101534