

# (Real2Sim)<sup>-1</sup>: 3D Branch Point Cloud Completion for Robotic Pruning in Apple Orchards

Tian Qiu<sup>1</sup>, Alan Zoubi<sup>2</sup>, Nikolai Spine<sup>3</sup>, Lailiang Cheng<sup>4</sup>, Yu Jiang<sup>5</sup>

**Abstract**—Robotic branch pruning, a rapidly growing field addressing labor shortages in agriculture, requires detailed perception of branch geometry and topology. However, point clouds obtained in agricultural settings often lack completeness, limiting pruning accuracy. This work addressed point cloud quality via a closed-loop approach, (Real2Sim)<sup>-1</sup>. Leveraging a Real-to-Simulation (Real2Sim) data generation pipeline, we generated simulated 3D apple trees based on realistically characterized apple tree information without manual parameterization. These 3D trees were used to train a simulation-based deep model that jointly performs point cloud completion and skeletonization on real-world partial branches, without extra real-world training. The Sim2Real qualitative results showed the model’s remarkable capability for geometry reconstruction and topology prediction. Additionally, we quantitatively evaluated the Sim2Real performance by comparing branch-level trait characterization errors using raw incomplete data and the best complete data. The Mean Absolute Error (MAE) reduced by 75% and 8% for branch diameter and branch angle estimation, respectively, which indicates the effectiveness of the Real2Sim data in a zero-shot generalization setting. The characterization improvements contributed to the precision and efficacy of robotic branch pruning.

## I. INTRODUCTION

Precision crop load management is crucial to the apple industry which holds significant economic importance in the United States and accounts for a farm value of 3.2 billion dollars in 2022 [1]. Pruning is the key to setting an optimal crop load potential for apple trees and directly influences fruit quality, tree vitality, and overall yield. Tree crop pruning is currently done by laborious manual operations, whereas the labor shortage, especially skilled workforce, not only increases the pruning cost but also reduces the pruning performance (e.g., consistency and precision) dramatically. In contrast, robotic pruning offers a solution to these challenges by leveraging computer vision and control technologies, providing consistent, efficient, and precise pruning that enhances overall orchard productivity and sustainability.

A few studies have investigated pruning robots for woody perennial crops such as apple [2], [3], cherry [4], and grape

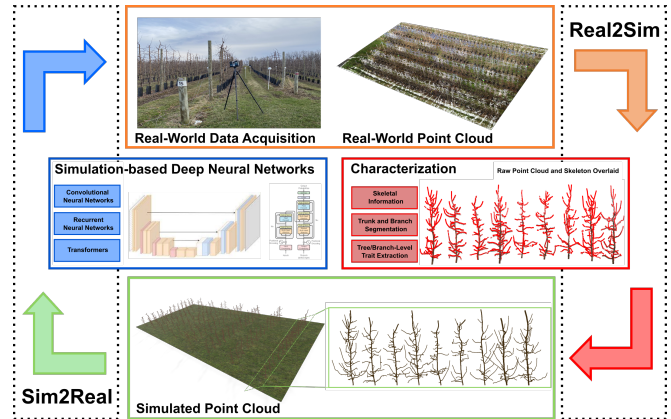


Fig. 1. The proposed (Real2Sim)<sup>-1</sup> (i.e., Real2Sim and Sim2Real) loop for deep learning models in the context of agriculture where large-scale datasets are usually unavailable and need enormous effort for development.

[5]. These robots were designed to use RGB-D cameras for obtaining local branch information, and subsequently prune branches or vines in a scan-and-cut manner [6]. For instance, a pruning robot can navigate to predefined waypoints and cut a branch once a target branch is identified. While this local pruning approach has demonstrated effectiveness given simple pruning rules (e.g., removal of side branches), apple trees require a global pruning strategy to reach optimal crop load potential [2]. This global pruning strategy considers both branch length and diameter for pruning. The length-based rule shortens branches that are longer than 40–45 cm to prevent the intervention to neighboring trees. The diameter-based rule uses both branch diameter and position (i.e., the height of a branch) to identify and remove the aged branches. Specifically, branches with a diameter exceeding 2 cm are selected as target branches. Subsequently, the largest target branch is pruned, followed by the highest one. Therefore, a global robotic pruning approach considering tree-wise branch distribution is urgently needed for apple crop load management.

A common method for the derivation of the global branch distribution is through tree quantitative structural modeling, which aims to reconstruct a quantitative structure model (QSM) of trees by capturing essential topology, geometry, and volume properties. The properties include the quantity, length, volume, angle, and size distribution of the branch. While QSM methods have shown effectiveness in various forest applications [7], [8], their application to apple tree

<sup>1</sup>Tian Qiu is with School of Electrical and Computer Engineering, Cornell University, Ithaca, USA tq42@cornell.edu

<sup>2</sup>Alan Zoubi is with Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, USA

<sup>3</sup>Nikolai Spine is with College of Arts&Sciences, Cornell University, Ithaca, USA

<sup>4</sup>Lailiang Cheng is with School of Integrative Plant Science, Cornell University, Ithaca, USA

<sup>5</sup>Yu Jiang is with School of Integrative Plant Science, Cornell University, Geneva, USA yujiang@cornell.edu

Code and Data

analysis faces significant challenges. The most prominent one is the inconsistency in QSM results arising from data quality issues [9]. Point clouds of apple trees acquired under field conditions suffer from incompleteness and discontinuity because of two primary factors: 1) modern orchards have high tree density that leads to high occlusion and low visibility to sensors and 2) adverse weather conditions such as variable sunlight and strong winds introduce noise and ghost points that further exacerbate data quality issues. Overcoming these challenges requires accurately reconstructing complete point clouds from incomplete raw data, so that branch information of individual trees can be used for global robotic pruning strategy.

In this study, we introduced the (Real2Sim)<sup>-1</sup> framework, a closed-loop methodology that centers around the Real2Sim data generation and Sim2Real application, aiming at providing a transformative solution for domains facing the data scarcity challenge to unlock the potential of learning-based approaches (Figure 1). To demonstrate the effectiveness of this loop, we applied it to address the prevalent data quality issues encountered in robotic pruning. Particularly, we utilized the Real2Sim data generation pipeline to produce simulated 3D apple tree models. These models were crafted from realistic apple tree geometric and topological data extracted by an apple tree characterization pipeline (referred to as **AppleQSM**) [9]. The simulated data was then used to train a joint completion and skeletonization model, achieving satisfactory completion results on real-world data in a zero-shot generalization setting. The Sim2Real capabilities showcased by the trained model underscored the efficacy of the closed-loop approach, both qualitatively and quantitatively, in addressing challenges in robotic pruning. Additionally, this framework opens opportunities for Digital Twins and Extended Reality by enabling real-time updates with high-fidelity simulated data, enhancing the accuracy and realism of virtual environments for and real-time control.

In summary, the major contribution of this study can be summarized in the following key points: 1) Unlike prior work that solely focused on Sim2Real, this is, to our best knowledge, the first study to propose a closed-loop approach in agriculture. This innovative method facilitates the adaptability and performance of learning-based approaches in domain areas, 2) the development of a joint transformer-based model coupled with the integration of a variance loss function, aimed at improving precision in predictions, and 3) the validation of the joint model’s Sim2Real capabilities, particularly in the context of robotic branch pruning, showcasing its efficacy in real-world applications.

## II. RELATED WORK

### A. 3D Tree Analysis

Analyzing trees using 3D computer vision techniques has been a long-lasting research area for forestry with a focus on tree inventory and biomass estimation [7], [8]. In agriculture, tree analysis provides critical information essential to making pruning decisions [10], [11]. Current pipelines for 3D tree analysis are broadly categorized into

geometry-based and learning-based approaches. A prominent tree modeling method within the geometry-based paradigm is QSM methods [12], [13]. Learning-based tree analysis approaches involve deep learning models to capture complex patterns and relationships in tree structures, providing more accurate and efficient analysis [14], [15]. These methods usually use deep neural models for the segmentation of raw point cloud data and process each segmented component separately to derive traits of interest.

### B. 3D Point Cloud Completion in Agriculture

Previous studies on point cloud completion for agricultural applications could be categorized into template-based and learning-based approaches. Template-based methods use fitting algorithms and determine the fitting template by the prior geometric properties of objects for the prediction of missing point clouds [16], [17]. However, template-based approaches heavily rely on the object geometry, and their optimal performance depends on extensive parameter tuning. In contrast, learning-based approaches leverage the strong expressiveness of deep neural networks to learn the geometric and topological features from incomplete objects and to complete missing parts. [18] proposed a deep learning-based pipeline for completing point clouds of plant leaves to estimate leaf area. [19] developed a deep neural network for tree completion to support robotic operations. Another line of work uses deep neural networks to learn the signed distance fields (SDF) for the completion of fruits [20], [21].

### C. Sim2Real Learning-based Models for Trees

Existing tree modeling studies fall into two categories: procedural and inverse procedural modeling. Procedural methods, like L-systems [23], [24], are powerful for simulating tree geometries but require manual parameter adjustment and domain expertise, leading to time-consuming and intricate processes. Conversely, inverse procedural techniques leverage experimental data (e.g., images and range scans), to automatically estimate parameters, offering a more accessible solution that allows a broader range of users to participate without requiring intricate procedural knowledge [25], [26].

The progress in tree modeling techniques has opened doors for domains with limited large-scale real-world tree data to utilize learning-based methods [27], [28], [29]. Nonetheless, a significant hurdle with simulation-based approaches lies in the disparity between simulated and real-world data, posing challenges for Sim2Real knowledge transfer. To address this gap, recent advances in domain adaptation techniques, as demonstrated by [30], aimed to enhance the Sim2Real performance of simulation-based models by leveraging knowledge learned from closely related simulation source domains. Additionally, domain randomization techniques have emerged as a popular approach to mitigate the domain gap by forcing networks to learn semantically relevant features that are invariant to superficial properties [31]. This is achieved through the introduction of variations in parameters, textures, and lighting conditions within the simulation environment. In agriculture, [27] pioneered the first attempt to fully train a

simulation-based deep model using domain randomization techniques. While their work showed promising Sim2Real branch segmentation performance across varying environments, a notable performance gap remained due to data distribution dissimilarity. Additionally, aligning semantically relevant features through domain randomization may not fully capture geometric and topological aspects.

### III. METHODOLOGY

#### A. Real2Sim Data Generation Pipeline

Current Sim2Real studies in agriculture, while effective, often rely on manually created simulation scenes, leading to unrealistic simulations [27], [28]. This is due to the complexity of tree growth processes influenced by internal signaling and external environmental and management factors such as the use of trellising training systems. The discrepancy between simulated and real-world data limits Sim2Real performance, particularly in zero-shot generalization scenarios. To address this, we introduced a Real2Sim data generator to produce accurate 3D apple tree models.

Since tree branches can be modeled as cylinders, our process began with the structural modeling of real-world tree point clouds using AppleQSM. AppleQSM extracted skeleton points and their associated radii, forming **skeletal spheres** which were used as input for the data generator. We developed a program using the Fusion 360 API (Autodesk, version 2.0.17453) that generated 3D tree models hierarchically, starting with the trunk and then growing branches. The process involved creating a 2D spline using the coordinates of the skeletal spheres, and then performing a sweep operation to construct a 3D cylindrical body. The sweep function incorporated a taper angle to simulate branching. In this study, the taper value was empirically set to -0.5 (negative for decreasing radius). This tapering approach was chosen over directly utilizing raw skeletal spheres' radii due to observed errors in skeletal radius towards branch ends, attributed to the increasing incompleteness of the point cloud. In addition to 3D meshes, we created 100 new skeleton points by evenly sampling from the spline to serve as the ground truth.

#### B. Joint Completion and Skeletonization Model

The joint model consists of a transformer encoder and a joint decoder (Figure 2). The encoder extracts rich general features from the input partial point cloud  $P_i \in \mathbb{R}^{K_i \times 3}$  and outputs point feature embedding and the coarse completion point cloud  $P_c \in \mathbb{R}^{K_c \times 3}$ . Employing the embedding and the coarse completion, the joint decoder generates respective geometric and topological features for heads to produce the refined completion point cloud  $P_r \in \mathbb{R}^{K \times 3}$  and the skeleton points  $P_s \in \mathbb{R}^{N \times 3}$ . Additionally, we formulated this joint model using the generative adversarial network (GAN) to enhance the global feature learning.

1) *Transformer Encoder*: We employed the concept of *point proxy* introduced in [32] to represent the input point cloud as a collection of these proxies. The transformer encoder [32], comprising geometry-aware self-attention modules, was used to capture the geometric relationships among

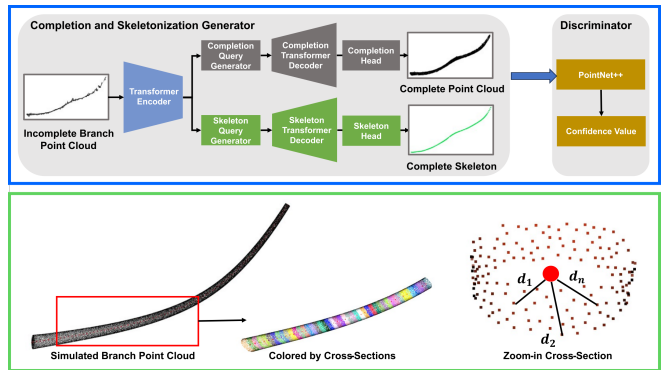


Fig. 2. Top: The developed joint completion and skeletonization model formulated as a GAN framework. Bottom: The illustration of the variance loss that aims to minimize the variance of the distance from surface points ( $d_1, d_2, \dots, d_n$ ) to the skeleton point to better constrain the geometry. The red dot represents the skeleton point for the cross-section.

the point proxies, ultimately generating the point query embedding  $Q'$ . The global features of these embeddings were used to produce the coarse completion results  $P_c$ .

$$Q' = \text{MLP}_e(E(P_i)), \quad P_c = \text{MLP}_{co}(M(Q')) \quad (1)$$

where  $E$  and  $M$  are the transformer encoder and the max-pooling operation, respectively.

2) *Joint Decoder*: Point queries were generated using the global features of the point query embedding along with the coarse completion point clouds, crucially serving as the initial state for the decoder. To facilitate the smooth optimization process in the joint setting, we developed two separate query generators to produce respective point queries for the initialization of completion and skeleton decoders ( $Q_c$  and  $Q_s$ ). Employing these separate point queries, the completion decoder proceeded to generate geometric features for the completion head to produce completion results, while the skeleton decoder generated topological features for its head to output skeleton points.

$$Q_c = \text{MLP}_c([M(Q'), P_c]), \quad Q_s = \text{MLP}_s([M(Q'), P_c]) \quad (2)$$

The incorporation of the skeleton head allowed the model to acquire essential topological features, thereby improving the quality of completion results, particularly in regions with discontinuities where pure geometric features may be insufficient. Furthermore, we integrated a skeleton loss to supervise the skeleton predictions, thereby reinforcing the constraints on geometry and topology, leading to more accurate and coherent predictions.

3) *Adversarial Discriminator*: Previous work has demonstrated that the adversarial learning strategy of the GAN framework can regularize the predictions from a global perspective and implicitly penalize outputs that deviate from the target [34]. Therefore, we formulated the joint model based on the GAN to enable the completion point generator to learn a richer variety of point distributions from the latent space. The discriminator aims to help implicitly evaluate the

completion points produced from the generator against the latent point distribution.

### C. Loss Function

1) *Geometry Loss*: The geometry loss measures the difference between the predicted completion point cloud and the ground truth point cloud. Since both point clouds are unordered, the loss needs to be invariant to the permutations of the points. We adopted Chamfer Distance as the completion loss for its  $O(N \log N)$  complexity [37] (Equation 3). Additionally, we used a repulsion loss  $L_R$  to obtain evenly distributed points [38].

$$L_{CD} = \frac{1}{|P_{gt}|} \sum_{p \in P_{gt}} \min_{c \in P_r} \|p - c\| + \frac{1}{|P_r|} \sum_{c \in P_r} \min_{p \in P_{gt}} \|c - p\| \quad (3)$$

2) *Variance Loss*: Branches can be conceptualized as cylindrical objects with small cross-sections (Figure 2). Cylindrical objects inherently exhibit an equal-distance property, where surface points maintain a consistent distance from points along the central axis. Leveraging this geometric prior, we proposed a novel variance loss  $L_V$  aimed at better constraining the geometry of predicted branch points. The variance loss seeks to minimize the variance of distances between ground truth skeleton points and predicted surface points (Equation 4). Here, the distance from a skeleton point to a surface point is defined as the minimum distance from the skeleton point to all surface points. To ensure a close match between skeleton points and surface points, it’s imperative that the completion predictions possess sound geometry. Therefore, we deferred the activation of this variance loss until the later fine-tuning stage.

$$L_V = \sum_{P=\{P_c, P_r\}} \text{Var} \left( \left\{ \min_{c \in P_{sgt}} \|p - c\|_2 \mid p \in P \right\} \right) \quad (4)$$

3) *Skeleton Loss*: We incorporated the unsupervised sampling loss [33] denoted by  $L_S$  to constrain the skeleton predictions in the first training stage. This sampling loss involves selecting points on the surface of each skeletal sphere and calculating the Chamfer Distance between these sampled points and the ground truth completion points. Furthermore, in the fine-tuning stage, we utilized the ground-truth skeleton points to supervise the joint model. This was feasible because we could easily generate ground truth data using the Real2Sim data generation pipeline. Specifically, we used the Chamfer Distance metric to compute the skeleton loss between predicted skeleton points and ground truth skeleton points  $P_{sgt}$ .

4) *Adversarial Loss*: To supervise the discriminator, we used the least square loss as the adversarial loss:

$$L_G = (D_{P_{gt}} - 1)^2, \quad L_D = (D_{P_{gt}})^2 + (D_{P_r} - 1)^2 \quad (5)$$

5) *Joint Loss*: We formulated the final joint loss (Equation 6) and used  $L_D$  to supervise the discriminator separately. We applied smaller weights to  $L_S$  and  $L_V$  to prioritize the learning of the completion task. It is worth noting that we used the variance loss  $L_V$  in the finetuning stage only because the existing completion datasets do not provide ground truth skeleton points.

$$L_J = L_{CD} + L_R + L_G + \lambda_1 L_S + \lambda_2 L_V \quad (6)$$

## IV. EXPERIMENT

### A. Dataset

To validate the efficacy of the proposed Real2Sim concept, we developed two distinct simulated datasets, namely FB (Fusion Branch) and NB (Nozeran Branch), to train the simulation-based joint model. Additionally, we utilized a real-world dataset, referred to as COB2022, collected by a terrestrial laser scanner (TLS) in an apple orchard, to evaluate the Sim2Real performance (Table I).

TABLE I  
SUMMARY OF DATASETS USED IN THIS STUDY.

| Dataset | #Branches | #Training Data | #Testing Data | Source |
|---------|-----------|----------------|---------------|--------|
| COB2022 | 106       | NA             | 106           | TLS    |
| NB      | 1432      | 1145           | 287           | L-Py   |
| FB      | 1432      | 1136           | 296           | CAD    |

1) *Real-World Field Dataset*: A total of 55 apple trees were used for data acquisition that was conducted during the off-season without any leaves causing occlusion to tree trunks and branches. These trees were planted at a spacing of 3.66 m (12 feet) by 0.91 m (36 inches) in 2011 and trained in the tall spindle system (Figure 1). We manually selected 106 branches from 9 apple trees and measured their branch diameter and angle using clipper and angle ruler, respectively, to form the COB2022 dataset.

2) *Simulated Dataset*: The rest of the 46 trees in the field dataset were used for the FB dataset generation using the Real2Sim data generation pipeline to make sure no data leakage occurs in model training. We adopted the same approach as [39] to produce point cloud data. On the other hand, we generated the NB dataset by creating randomized tree structures with varying branch cycles, axes, and radii based on the Nozeran tree growth model [40] in L-Py [41]. This model posits that trees grow by repeated iteration of a basic unit of structure, known as a “tree unit”. The tree unit consists of a central leader from which lateral branches emerge following a certain pattern, which is similar to the overall apple tree structure.

### B. Model Training

We leveraged the transfer learning strategy to train the simulation-based branch point cloud joint model in an end-to-end manner with a two-stage training procedure. As transformer-based models need large-scale datasets for pre-training to learn rich contextual representations and semantic

relationships, we used the PCN dataset [39] to train the joint model using the joint loss  $L_J$  without the variance loss  $L_V$  at Stage 1. Next, we finetuned the pretrained joint model on the simulated dataset (NB or FB dataset) for efficient adaptation to domain-specific nuances by turning on variance loss  $L_V$  at Stage 2.

The joint model was implemented using PyTorch [43]. The skeleton loss weight  $\lambda_1$  was set to 0.01 at both stages, whereas the variance loss weight  $\lambda_2$  was set to 10 whenever it was turned on. We trained the joint model for 384 epochs with a batch size of 16 at Stage 1 and for 800 epochs with a batch size of 64 at Stage 2. More technical details for the transformer-based encoder and decoder can be found in [32].

### C. Simulation-based Model Evaluation

We conducted the assessment to evaluate the completion performance of simulation-based models. This involved visual assessment of the completion results and quantitative comparison on COB2022 using AppleQSM that relies on geometric features of input tree point clouds to identify architectural traits. As a result, by computing the error in trait characterization for various completion models, we could accurately quantify the completion performance. It's noteworthy that while the simulation-based model achieved completion at the branch level, AppleQSM necessitated tree-level input data. Consequently, we manually segmented all primary branches for 9 apple trees in the COB2022 dataset.

1) *Completion Evaluation:* For the completion evaluation, we conducted a qualitative assessment of the completion results generated by various models, with a primary focus on the fidelity of geometry and topology reconstruction. As a benchmark, we employed AdaPoinTr [32], a state-of-the-art point cloud completion model, and subsequently trained four distinct models to investigate the influence of simulated datasets and model architecture. These models included AdaPoinTr-NB (AdaPoinTr trained on the NB dataset), and AdaPoinTr-FB, which aimed to underscore the efficacy of the Real2Sim dataset. In addition to it, we trained the developed joint model on the FB dataset using different loss functions to elucidate the merits of the joint design and variance loss integration. For simplicity, we referred to the joint model trained by necessary losses ( $L_{CD}$ ,  $L_R$ ,  $L_G$ , and  $L_S$ ) as Joint-GS and the joint model trained by adding another variance loss as Joint-GSV.

Additionally, we adopted the mean  $CD-l_1$  (i.e., the  $L_1$  version of Chamfer Distance) as the metric for the quantitative evaluation of the completion performance following existing works [32], [39]. We conducted the evaluation only using the FB dataset because it provides significantly more realistic geometric and topological features than the NB dataset, thereby the quantitative evaluation is more meaningful.

2) *Characterization Evaluation:* We used the complete branch point clouds from all four models for the tree assembly, resulting in four complete tree point clouds in addition to the raw tree point cloud. These tree data were characterized by AppleQSM for the estimation of branch diameter and branch angle. By comparing the AppleQSM measurements

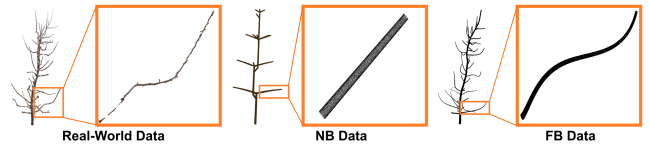


Fig. 3. Branch representative from each dataset. The real-world data presents significant incompleteness and discontinuity, which is the major bottleneck for robotic perception. The NB data shows simple geometry and lacks intricate structures. The FB data shows significantly more realistic geometry and organic structure.

with ground truth measurements in the COB2022 dataset, we computed metrics including mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). Through these evaluations, we aimed to ascertain the efficacy and contributions of the Real2Sim dataset, the joint model, and the novel variance loss toward accurate tree characterization, which is crucial for informing robotic pruning strategies effectively.

## V. RESULT AND DISCUSSION

### A. Real2Sim Data Generation

The developed Real2Sim data generation presented the capability to produce a realistic representation of apple trees with intricate organic structures, indicating the effectiveness of the modeling process (Figure 3). Modeling trees presents a significant challenge due to their growth patterns being influenced by a mix of internal factors such as hormones, as well as external factors like gravitropism and phototropism. While existing computer graphics techniques for tree geometry generation have reached a level of maturity that allows for the creation of detailed 3D models, they often require a certain level of expertise to operate effectively. Furthermore, these techniques primarily focus on modeling forest and urban trees, which exhibit remarkably different structures compared to fruit trees such as apple trees. Notably, there is currently no publicly available dataset comprising realistic representations of apple trees suitable for training deep completion models. Through the utilization of the Real2Sim data generation pipeline, we have seamlessly bridged simulated data with real-world field data, all without the need for manual intervention, thereby ensuring the efficiency and realism of the simulation (Figure 1).

Furthermore, the  $(\text{Real2Sim})^{-1}$  approach holds significant promise for the agricultural robotics community, offering a transformative toolset with broad-reaching implications. By facilitating the creation of highly realistic simulations, this pipeline opens new avenues for the development and validation of learning-based robotic systems tailored specifically for various domains. The adaptability of this pipeline extends beyond apple trees, offering a versatile solution applicable to a variety of crops and tree species. This adaptability addresses the critical issue of data scarcity, providing a valuable resource for researchers working across diverse science domains.

## B. Sim2Real Completion Result

The AdaPoinTr-NB model exhibited a tendency to fit straight-up shapes for input partial branches, lacking the capability to accurately represent curvature geometry. This limitation stemmed from the inadequately realistic data representations in the NB dataset (Figure 4). Moreover, this model over-completed the curvature areas by adapting the straight-up geometries, which could introduce significant errors into subsequent characterization. In contrast, AdaPoinTr-FB demonstrated notably superior performance compared to AdaPoinTr-NB, offering more biologically plausible completion results. This enhancement was attributed to the improved realism provided by the FB dataset, which accurately modeled the distribution of real-world branch data and enabled the deep completion model to learn rich and realistic geometric features. However, AdaPoinTr-FB faced challenges in accurately reconstructing raw branch points around trunk-branch junction areas, crucial for precise branch-level trait extraction, particularly branch diameter estimation. Specifically, AdaPoinTr-FB provided misaligned completion results where the raw points located inside the complete points (Blue box in Figure 4) and completely missed one of the disconnected branch segments due to confusion caused by gaps (Green box in Figure 4). These completion errors constrained downstream characterization accuracy and pruning optimality.

The joint model (i.e., Joint-GS and Joint-GSV) generated better alignment and resolved disconnected segments more faithfully, implying the efficacy of the GAN framework and the joint design. The adversarial training strategy facilitates global information learning and the joint design bridges the completion head and the skeleton head thereby the skeleton predictions could provide complementary topological features for the completion. This performance improvement was confirmed by the  $CD-l_1$  ( $\times 1000$ ) metric where Joint-GS and Joint-GSV achieved a respective 1.118 and 0.980, outperforming the 1.131 from AdaPoinTr-FB. Furthermore, the variance loss improved the completion results qualitatively and quantitatively by encouraging more accurately filled-in points and smoother surfaces with lower variability (Figure 4). Without using the variance loss, the model presented relatively rough surfaces that were slightly deviated from the correct geometry due to the limitations of the Chamfer Distance loss  $L_{CD}$  that could not capture all aspects of the geometric relationship given highly irregular partial branch point cloud. In contrast, the variance loss enhanced the geometric relationship by leveraging the geometric prior of cylindrical shape, providing a more coherent completion.

## C. Sim2Real Characterization Result

The evaluation of completion models using AppleQSM revealed intriguing insights into the impact of simulated datasets and model architecture on the downstream characterization performance (Table II). Despite the inherent limitations of the NB dataset, derived from the Nozeran tree growth model and characterized by simplistic branch representations, AdaPoinTr-NB demonstrated a degree of

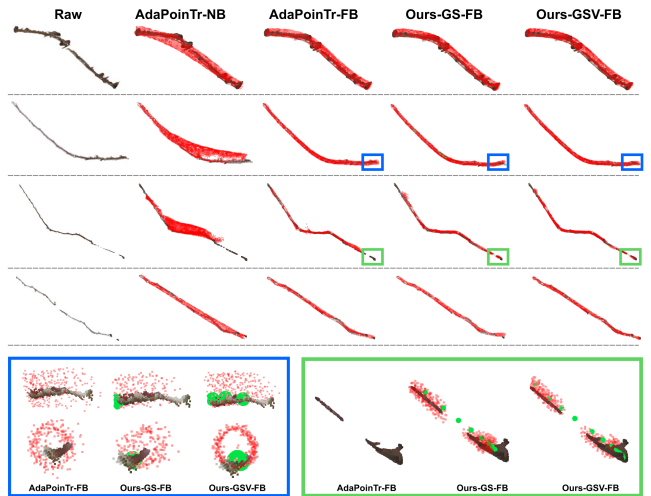


Fig. 4. Completion results generated by different models. The red points represent the complete points while the green points are the skeleton points. The first two rows show the partial branches while the last two rows show the partial and discontinued branches. The bottom blue and green boxes present the zoom-in completion results of the corresponding selected area in the second and third rows. We referred to the joint model as ours. GS stands for generative and skeleton losses and V for variance loss.

TABLE II

BRANCH-LEVEL TRAIT ESTIMATION ERROR FROM DIFFERENT MODELS. WE REFERRED TO THE JOINT MODEL AS OURS. GS STANDS FOR GENERATIVE AND SKELETON LOSSES AND V FOR VARIANCE LOSS.

| Model     | Dataset | Branch Diameter |              |             | Branch Angle |             |             |
|-----------|---------|-----------------|--------------|-------------|--------------|-------------|-------------|
|           |         | MAE             | MAPE         | RMSE        | MAE          | MAPE        | RMSE        |
| Baseline  | NA*     | 7.11            | 47.94        | 7.43        | 7.46         | 10.11       | 11.22       |
| AdaPoinTr | NB      | 3.43            | 24.05        | 4.40        | 9.32         | 12.73       | 13.30       |
| AdaPoinTr | FB      | 2.35            | 15.22        | 3.62        | 7.29         | 10.19       | 9.50        |
| Ours-GS   | FB      | 2.12            | 14.63        | 2.77        | 8.14         | 11.17       | 12.31       |
| Ours-GSV  | FB      | <b>1.77</b>     | <b>12.27</b> | <b>2.28</b> | <b>6.84</b>  | <b>9.42</b> | <b>9.16</b> |

\* The baseline model is a geometry-based model and does not require training.

The unit of MAE and RMSE is  $mm$  and that of MAPE is  $\%$ .

efficacy, reducing the MAE of the branch diameter estimation by more than 50%. This underscored the robustness of the baseline completion model (i.e., AdaPoinTr) and highlighted the importance of utilizing simulated datasets with high domain similarity. Conversely, the utilization of the more realistic FB dataset in AdaPoinTr-FB resulted in a substantial reduction in MAE of the branch diameter estimation, close to 70%. This significant Sim2Real improvement underscored the crucial role played by the Real2Sim dataset in enhancing completion performance. The increased realism provided by the FB dataset enables the model to learn richer features, leading to more accurate completion results. Joint-GS demonstrated superior performance compared to AdaPoinTr-FB, leveraging the benefits of both the GAN framework and the joint setting highlighted in the completion evaluation. Building upon the success of Joint-GS, Joint-GSV achieved even greater accuracy by using the variance loss as an enhanced geometry regularization.

The characterization results obtained from Joint-GSV accurately indicated the aged branches required pruning, which

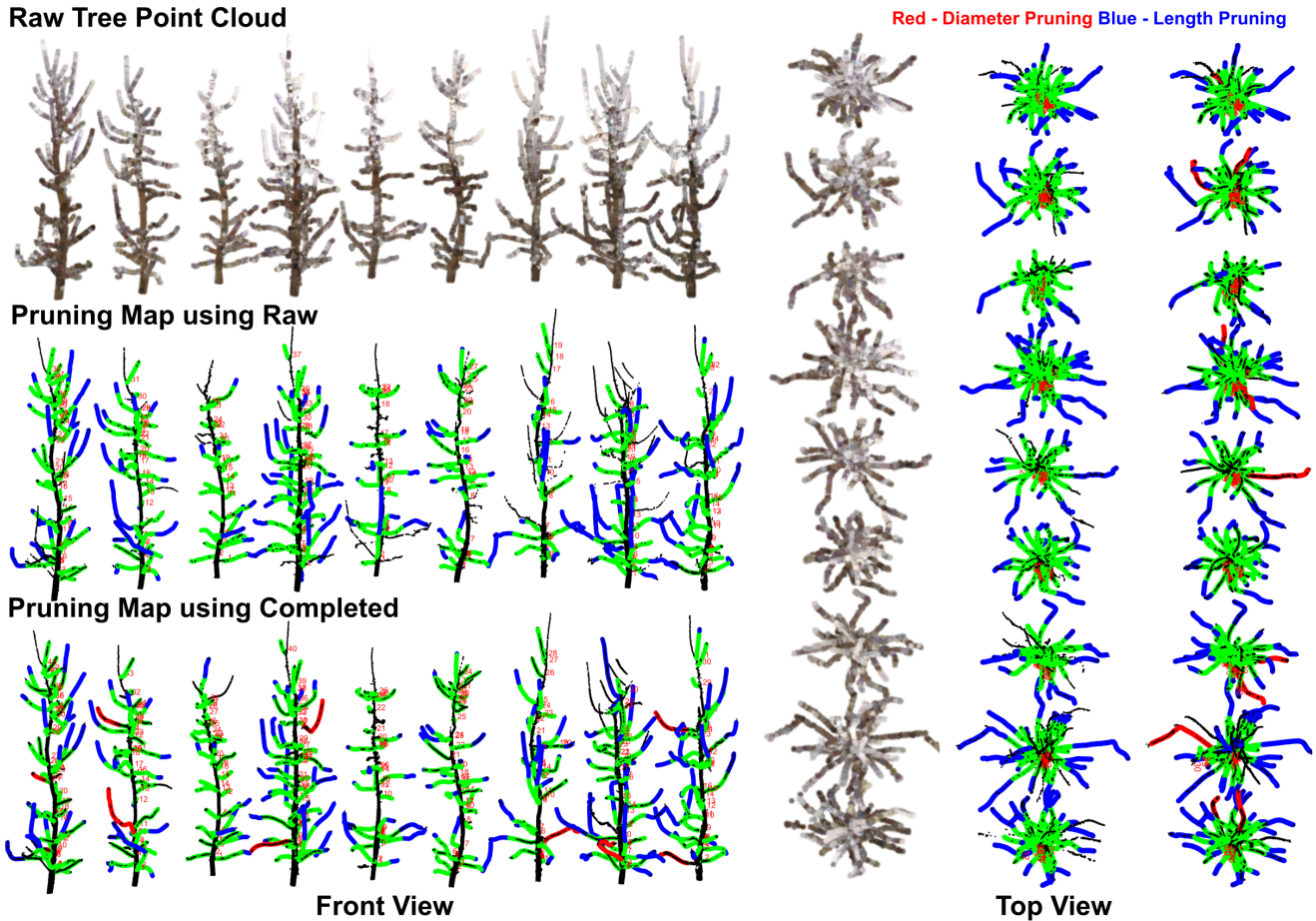


Fig. 5. Pruning map developed based on the characterization results using raw data and the complete data from Ours-GSV-FB.

was pivotal in guiding pruning robots to maintain the tree productivity in the context of apple crop load management. We generated pruning maps based on these characterization results that provide valuable insights into optimizing pruning decisions to effectively manage crop load (Figure 5). In contrast to raw data characterization results, which suffered from underestimation of branch diameter due to incomplete branch point cloud data, Joint-GSV significantly improved diameter estimation accuracy. This enhancement ensured that branches are accurately classified based on their position, diameter, and length, allowing for more precise identification of branches that require pruning. The importance of this precision becomes apparent when considering the outcomes of incorrect branch pruning on crop load management. Failing to accurately identify and prune branches exceeding the diameter cutoff threshold can lead to overcrowded tree canopies and inefficient resource allocation, resulting in sub-optimal fruit development and reduced yield and quality. By leveraging the improved characterization accuracy provided by Joint-GSV, the pruning map offers a more nuanced understanding of the orchard’s spatial distribution. This enables targeted pruning interventions where necessary, contributing to more efficient and effective crop load management. Thus, the advancements made in this study could revolutionize the

field of robotic pruning and have far-reaching impacts on agricultural practices.

## VI. CONCLUSIONS

A simulation-based deep completion network with a joint completion and skeletonization design was developed for robotic pruning in apple orchards. While the joint model was trained on simulated datasets only, its capability of generating highly accurate complete branches from real-world partial branch offered significantly enhanced geometric and topological features to enable AppleQSM to accurately characterize branch-level traits. This achievement paved the way for more accurate and efficient robotic pruning strategies. The success of the joint model’s Sim2Real process owes much to the innovative Real2Sim data generation pipeline, which streamlines the creation of highly realistic apple tree models without requiring manual intervention. Moreover, we conceptualized the entire process as  $(\text{Real2Sim})^{-1}$ , identifying it as a crucial component with promising potential for broader adoption across various domains. This loop addresses the challenge of domain-specific data scarcity by seamlessly bridging the gap between simulated and real-world data sources.

In future studies, we will focus on 1) training the joint

model directly on tree point clouds to improve efficiency and leverage global tree information and 2) transferring the pruning map to robots for real-world pruning operations.

## ACKNOWLEDGMENT

The present study was supported by the USDA NIFA Hatch project (accession No. 1025032) and USDA NIFA Specialty Crop Research Initiative (award No. 2020-51181-32197). The authors would gratefully thank Kaspar Kuehn for helping measure the branch diameter and branch angle for trees used in this study.

## REFERENCES

- [1] USApple. *Industry at a Glance*. 2022. Available online: <https://usapple.org/industry-at-a-glance>.
- [2] He, Long, and James Schupp. "Sensing and automation in pruning of apple trees: A review." *Agronomy* 8.10 (2018): 211.
- [3] Zahid, Azlan, et al. "Technological advancements towards developing a robotic pruner for apple trees: A review." *Computers and Electronics in Agriculture* 189 (2021): 106383.
- [4] You, Alexander, et al. "An autonomous robot for pruning modern, planar fruit trees." *arXiv preprint arXiv:2206.07201* (2022).
- [5] Silwal, Abhisesh, et al. "Bumblebee: A path towards fully autonomous robotic vine pruning." *arXiv preprint arXiv:2112.00291* (2021).
- [6] You, Alexander, et al. "An efficient planning and control framework for pruning fruit trees." 2020 IEEE international conference on robotics and automation (ICRA). IEEE, 2020.
- [7] Bornand, Aline, et al. "Individual tree volume estimation with terrestrial laser scanning: Evaluating reconstructive and allometric approaches." *Agricultural and Forest Meteorology* 341 (2023): 109654.
- [8] Allen, Matthew J., et al. "Tree species classification from complex laser scanning data in Mediterranean forests using deep learning." *Methods in Ecology and Evolution* 14.7 (2023): 1657-1667.
- [9] Qiu, Tian, et al. "AppleQSM: Geometry-Based 3D Characterization of Apple Tree Architecture in Orchards." *Plant Phenomics* 6 (2024): 0179.
- [10] Akbar, Shayan A., Noha M. Elfiky, and Avinash Kak. "A novel framework for modeling dormant apple trees using single depth image for robotic pruning application." 2016 IEEE international conference on robotics and automation (ICRA). IEEE, 2016.
- [11] Tabb, Amy, and Henry Medeiros. "A robotic vision system to measure tree traits." 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017.
- [12] Raunonen, Pasi, et al. "Fast automatic precision tree models from terrestrial laser scanner data." *Remote Sensing* 5.2 (2013): 491-520.
- [13] Fan, Guangpeng, et al. "AdQSM: A new method for estimating above-ground biomass from TLS point clouds." *Remote Sensing* 12.18 (2020): 3089.
- [14] Wilkes, Phil, et al. "TLS2trees: A scalable tree segmentation pipeline for TLS data." *Methods in Ecology and Evolution* 14.12 (2023): 3083-3099.
- [15] Dong, Xuhua, et al. "Three-dimensional quantification of apple phenotypic traits based on deep learning instance segmentation." *Computers and Electronics in Agriculture* 212 (2023): 108156.
- [16] Gené-Mola, Jordi, et al. "In-field apple size estimation using photogrammetry-derived 3D point clouds: Comparison of 4 different methods considering fruit occlusions." *Computers and electronics in agriculture* 188 (2021): 106343.
- [17] Marks, Elias, Federico Magistri, and Cyrill Stachniss. "Precise 3D reconstruction of plants from UAV imagery combining bundle adjustment and template matching." 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022.
- [18] Chen, Haibo, et al. "Point Cloud Completion of Plant Leaves under Occlusion Conditions Based on Deep Learning." *Plant Phenomics* 5 (2023): 0117.
- [19] Xu, Dali, Guangsheng Chen, and Weipeng Jing. "A Single-Tree Point Cloud Completion Approach of Feature Fusion for Agricultural Robots." *Electronics* 12.6 (2023): 1296.
- [20] Pan, Yue, et al. "Panoptic Mapping with Fruit Completion and Pose Estimation for Horticultural Robots." *arXiv preprint arXiv:2303.08923* (2023).
- [21] Magistri, Federico, et al. "Contrastive 3D shape completion and reconstruction for agricultural robots using RGB-D frames." *IEEE Robotics and Automation Letters* 7.4 (2022): 10120-10127.
- [22] Zhang, Chengjian, et al. "Apple tree branch information extraction from terrestrial laser scanning and backpack-lidar." *Remote Sensing* 12.21 (2020): 3592.
- [23] Prusinkiewicz, Przemyslaw. "Graphical applications of L-systems." *Proceedings of graphics interface*. Vol. 86. No. 86. 1986.
- [24] Prusinkiewicz, Przemyslaw, and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.
- [25] Wang, Guan, et al. "Statistical modeling of the 3D geometry and topology of botanical trees." *Computer graphics forum*. Vol. 37. No. 5. 2018.
- [26] Lee, Jae Joong, Bosheng Li, and Bedrich Benes. "Latent L-systems: Transformer-based Tree Generator." *ACM Transactions on Graphics* 43.1 (2023): 1-16.
- [27] You, Alexander, Cindy Grimm, and Joseph R. Davidson. "Optical flow-based branch segmentation for complex orchard environments." 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022.
- [28] You, Alexander, et al. "Semiautonomous Precision Pruning of Upright Fruiting Offshoot Orchard Systems: An Integrated Approach." *IEEE Robotics & Automation Magazine* (2023).
- [29] Bryson, Mitch, Feiyu Wang, and James Allworth. "Using Synthetic Tree Data in Deep Learning-Based Tree Segmentation Using LiDAR Point Clouds." *Remote Sensing* 15.9 (2023): 2380.
- [30] Wang, Mei, and Weihong Deng. "Deep visual domain adaptation: A survey." *Neurocomputing* 312 (2018): 135-153.
- [31] James, Stephen, et al. "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
- [32] Yu, Xumin, et al. "AdaPoinTr: Diverse Point Cloud Completion with Adaptive Geometry-Aware Transformers." *arXiv preprint arXiv:2301.04545* (2023).
- [33] Lin, Cheng, et al. "Point2skeleton: Learning skeletal representations from point clouds." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
- [34] Li, Ruihui, et al. "Pu-gan: a point cloud upsampling adversarial network." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
- [35] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [36] Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." *Advances in neural information processing systems* 30 (2017).
- [37] Fan, Haoqiang, Hao Su, and Leonidas J. Guibas. "A point set generation network for 3d object reconstruction from a single image." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [38] Yu, Lequan, et al. "Pu-net: Point cloud upsampling network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [39] Yuan, Wentao, et al. "Pcn: Point completion network." 2018 international conference on 3D vision (3DV). IEEE, 2018.
- [40] Tomlinson, Philip B. "Tree architecture: new approaches help to define the elusive biological property of tree form." *American Scientist* 71.2 (1983): 141-149.
- [41] Boudon, Frédéric, et al. "L-Py: an L-system simulation framework for modeling plant architecture development based on a dynamic language." *Frontiers in plant science* 3 (2012): 76.
- [42] Loshchilov, Ilya, and Frank Hutter. "Fixing weight decay regularization in adam." (2018).
- [43] Paszke, Adam, et al. "Pytorch: An imperative style, high-performance deep learning library." *Advances in neural information processing systems* 32 (2019).