Plant Phenomics 8 (2026) 100240

Contents lists available at ScienceDirect

Plant Phenomics

journal homepage: www.sciencedirect.com/journal/plant-phenomics

Research Article

A-Occ-Plant: Plant occluded point cloud completion via amodal segmentation

Jae Joong Lee, Bedrich Benes *

Department of Computer Science, Purdue University, 305 N University Street, West Lafayette, Indiana, 47906, USA

ARTICLE INFO

Keywords:
Point cloud completion
Amodal segmentation
Deep learning

ABSTRACT

Plants are geometrically and topologically complex objects, and methods and devices that produce plant point clouds often miss parts due to self occlusions, making further analysis, such as phenotypic trait extraction or 3D reconstruction, difficult. We introduce *A-Occ-Plant*, a novel method for point cloud completion. The first novelty of our algorithm is converting point clouds into a set of images, which are then completed using 2D amodal segmentation. The images are then converted into a complete point cloud by using view-consistent Gaussian splats. The second novelty is the use of a coarse-to-fine hierarchical Transformer with cross-scale attention. The completed soft masks are fused into a continuous 3D density field using Gaussian splatting, removing the need for external pose estimation or fixed-size inputs. We introduce a synthetic dataset using a procedural model and a real-world plant reconstruction benchmark with artificially generated occlusions. We further benchmark *A-Occ-Plant* against representative 3D point-cloud completion methods, demonstrate that it recovers downstream phenotypic traits (leaf count, leaf angle, plant height), and show that it generalizes to another crops (soybean). A-Occ-Plant achieves a 264.8% improvement in LPIPS and an 8.3% gain in SSIM compared to the current state of the art, while using only 2.3% of the parameters and running 39.4× faster. We release our code at https://github.com/JaeLee18/PlantPhenomics_Occlusion.

1. Introduction

1.1. Problem and motivation

While the field of human-made 3D object reconstruction has seen significant advances and is nearing maturity Huang et al. [1], plants are geometrically and topologically complex structures, and their capture and reconstruction remain challenging problems. One of the key tasks in vegetation capture is trait extraction, for example, branching angle, phyllotaxis, or leaf area index. The extracted traits play an essential role in many areas, such as leaf counting Ostermann et al. [2], digital twins creation Lee et al. [3]; Vazquez et al. [4], high-throughput plant phenotyping Yang et al. [5], and environmental monitoring Himeur et al. [6]. However, these methods require large-scale data capture, and manual approaches are error-prone and do not scale. The 3D scanning of vegetation into point clouds is the most common approach, but it must deal with dense foliage, overlapping branches, and limited sensor viewpoints, which lead to unavoidable occlusions and large areas of missing data.

The state of the methods for point cloud completion uses neural networks Yuan et al. [7]; Yu et al. [8]; Xia et al. [9]; Rong et al. [10], which learn to predict missing points from partial inputs. These methods work well for flat areas, symmetries, or areas with constant curvature. However, they do not perform well on stochastic structures. They often require a fixed number of input points (e.g. 1024 or 4096), and random subsampling of higher-resolution scans discards critical geometric details common in vegetation. Due to this limitation, the models cannot exploit the variable density of real-world scans.

Our key observation is that amodal segmentation in the image domain has seen rapid progress due to Transformer-based Tran et al. [11]; Gao et al. [12] and diffusion-based models Ozguroglu et al. [13]; Zhan et al. [14]; Xu et al. [15]; Lee et al. [16]. While these methods can convincingly fill missing regions in single views, they do not enforce consistency across multiple viewpoints, which limits their use for full 3D reconstruction.

* Corresponding author.
E-mail addresses: lee2161@purdue.edu (J.J. Lee), bbenes@purdue.edu (B. Benes).

https://doi.org/10.1016/j.plaphe.2026.100240
Received 24 March 2026; Received in revised form 27 May 2026; Accepted 11 June 2026
Available online 23 June 2026

1.2. Related work

Our method relates to 3D point cloud completion and 2D amodal segmentation.

1.2.1. 3D point-cloud completion

Early works on point-cloud completion operate directly in 3D, either by learning a mapping from sparse inputs to dense outputs or by deforming template shapes. Notable methods include Point Completion Network (PCN) Yuan et al. [7], which uses an encoder–decoder to predict coarse-to-fine point sets; TopNet Tchapmi et al. [17], which exploits a tree structure for generation; and GRNet Xie et al. [18], which lifts 2D depth to 3D via volumetric grids. More recent methods include the SnowflakeNet Xiang et al. [19], which introduces Snowflake Point Deconvolution for capturing detailed local geometry; PoinTr Yu et al. [8], which employs geometry-aware Transformers for set-to-set completion; PMP-Net++ Wen et al. [20], which enhances multi-step point moving with transformer attention; EINet Cai et al. [21], which reformulates completion as feature-space extrapolation and interpolation to capture fine details; and ComPC Huang et al. [22], which leverages pre-trained 2D diffusion priors to generate missing point information. Other methods Zhou et al. [23]; Zarei et al. [24]; Liu et al. [25] detect parts of plants that are then connected topologically. However, these methods rely on fixed input dimensionality, which requires the number of input points to match the training distribution, and thus struggle to handle the variable densities and high-resolution scans encountered in real-world plant data. Further, operating directly in 3D limits the computational efficiency and flexibility of the image-space approaches. Our approach, leveraging amodal segmentation in 2D space with Gaussian splatting Kerbl et al. [26], removes this hard limitation of fixed dimensionality. Moreover, we compare our approach with three point cloud completion methods Wang et al. [27]; Yu et al. [28]; Chen et al. [29] on the reconstruction metrics and downstream phenotypic traits.

1.2.2. Amodal segmentation

It aims to infer the complete extent of objects, including regions that are partially or fully occluded in the input image. Early methods extended instance segmentation frameworks such as Mask R-CNN by incorporating shape priors and visibility cues Zhu et al. [30]; Ehsani et al. [31]; Zhang et al. [32]; Zhan et al. [33]; Xiao et al. [34]; Tran et al. [11]. More recently, diffusion-based approaches have demonstrated state-of-the-art performance. Notably, pix2gestalt Ozguroglu et al. [13] and Amodal in the Wild Zhan et al. [14] achieve strong results on challenging benchmarks, including COCO-A Zhu et al. [35] and BSDS-A Zhu et al. [36]. Complementary efforts have also explored zero-shot foundation models such as SAM for amodal segmentation Liu et al. [37], as well as temporal extensions to video using diffusion priors Chen et al. [38].

We adopt pix2gestalt Ozguroglu et al. [13] as our baseline due to its publicly available training code and strong performance on complex, articulated objects. Although Amodal in the Wild Zhan et al. [14] achieves competitive results, its closed-source implementation limits reproducibility and prevents direct integration. Unlike prior works, our method performs amodal segmentation across diverse viewpoints, enabling consistent multi-view reasoning for 3D reconstruction.

1.3. Contributions

Building on the success of amodal segmentation in images, we propose *A-Occ-Plant*, a novel method to complete vegetation point clouds from heavily occluded 3D scans. *A-Occ-Plant* takes the incomplete point cloud and renders it into hundreds or thousands of views, each with carefully calibrated camera location and orientation. We then complete each view with a coarse-to-fine hierarchical Transformer and fuse the resulting masks into a continuous 3D density field using Gaussian splatting Kerbl et al. [26]. We will share the code and model, and our main contributions are:

1. Conversion of the problem of 3D point cloud completion into a set of 2D image completions using amodal segmentation.
2. A artificial occlusion generation pipeline and benchmark dataset designed to reflect real-world scanning artifacts in both controlled and in-field plant scenarios.
3. A coarse-to-fine hierarchical Transformer architecture that captures global context and fine structure through cross-scale attention for amodal mask completion.
4. Two paired datasets of complete and occluded maize and sorghum point clouds, together with their multi-view rendered masks and camera parameters, enabling reproducible evaluation of 2D amodal completion and 3D reconstruction.
5. A direct comparison against representative learning-based 3D point-cloud completion methods on both reconstruction metrics and downstream phenotypic traits.
6. A downstream phenotyping study showing that our completion produces phenotypically more accurate reconstructions (leaf count, leaf angle, and plant height), together with a generalization study to a second crop, soybean Sun et al. [39], with a distinct morphology.

2. Method

A-Occ-Plant begins by rendering multi-view masks from the scanned 3D point clouds as shown in Fig. 1. This data is then used to train a hierarchical transformer for amodal completion.

2.1. Preliminaries

We introduce the notations for our pipeline: rendering, 2D amodal segmentation, and Gaussian splatting Kerbl et al. [26]. Note that we need complete point clouds for testing and validation. We generate them by sampling virtual plants and from manually scanned real plants as described in Sect. 2.5.

2.2. Rendering

Let

$$\mathscr{P}_{\text{full}} = \{\mathbf{p}_i^{full}\}_{i=1}^{N_{full}},$$

denote the *full* (ground-truth) point cloud, where $\mathbf{p}_i^{\text{full}}$ are the 3D points, and

$$\mathscr{P}_{\text{occl}} = \{\mathbf{p}_i^{\text{occl}}\}_{i=1}^{N_{occl}},$$

where $\mathbf{p}_i^{\text{occl}}$ are the 3D points of the *occluded* (partial) point cloud.

We render these into M views using camera extrinsics ($\mathbf{R}_m$, $\mathbf{t}_m$) (orientation and position) and intrinsics $\mathbf{K}_{\text{int}} \in \mathbb{R}^{3\times3}$.

We distinguish between masks S_{full} rendered from the complete point cloud $\mathscr{P}_{\text{full}}$ and those S_{occl} rendered from the occluded point cloud $\mathscr{P}_{\text{occl}}$.

2.3. 2D amodal segmentation

We then apply a 2D amodal segmentation network

$$f_\theta : \{0,1\}^{H\times W} \to (0,1)^{H\times W},$$

which outputs per-pixel probabilities via a sigmoid. Denoting

$$\widehat{S}_{full}(u,v) = f_\theta(S_{occl})(u,v),$$

we interpret $\widehat{S}_{full}(u,v) \in (0,1)$ as the network's confidence that pixel (u, v) belongs to the full point cloud, which is similar to a soft mask instead of a thresholding binary mask.

2.4. Gaussian Splatting

Given the completed soft masks $\widehat{S}_{full}(u,v)$ and the known intrinsics $\mathbf{K}_{\text{int}}$ and extrinsics $(\mathbf{R}_m, \mathbf{t}_m)$, we use the Gaussian splatting Kerbl et al. [26] with the default parameter settings. The final point cloud is denoted by $\mathscr{P}_{\text{compl}}$ and reconstructed by extracting points corresponding to the highest-density regions in the continuous field.

2.5. Data generation and preprocessing

We use two datasets to train our model: 1046 field-grown maize plants scanned via Terrestrial Laser Scanning (TLS) Kimara et al. [40] and 1000 realistic 3D procedural sorghum plant models Ostermann et al. [2]. Each point cloud has approximately 10,000 points. We construct a dataset of paired occluded and complete amodal masks derived from high-fidelity 3D point clouds.

Each point cloud $\mathscr{P}_{occl}$ is canonicalized by translating its centroid $\mathbf{c}$ to the origin and scaling it uniformly such that the object fits within a unit sphere. This normalization step standardizes scale and position across all objects, ensuring consistent rendering and helping model generalization.

We then generate camera poses by sampling points on the surface of a sphere that encloses the normalized object. The sphere radius r_{cam} is set to be $1.5\times$ the object's normalized scale. To ensure comprehensive coverage, we tessellate the sphere into 200 regularly geodesically spaced points, position the camera at each point, and set the viewing direction to the coordinate system's center (see Fig. 9 for the effect of the number of views on the point cloud's completeness quality).

From each camera pose, we render a 2D image as a $W \times H = 128 \times 128$ binary mask of the complete, normalized object. This rendered image (see GT images in Fig. 3), which shows the object from that specific viewpoint, serves as the ground truth amodal mask S_{full}. The input to our network, the partial mask S_{occl}, is then created by synthetically occluding this ground truth view (e.g., using masks from other objects or geometric primitives). The final dataset consists of pairs (S_{occl}, S_{full}). We also store camera extrinsics and intrinsics for Gaussian Splatting.

To simulate realistic occlusion patterns in natural plant scans, we apply a randomized sequence of occlusion operations to each normalized point cloud $\mathscr{P}$. Specifically, we invoke a random subset of ten occlusion operations drawn from a pool of three routines. This diversity in occlusion patterns enables robust learning and generalization to real-world scenarios.

(1) **Viewpoint-based Ray Casting**

We simulate occlusion from a scanning device placed at position $\mathbf{v}$, either aligned with the object's principal axis or chosen randomly. The input point cloud $\mathscr{P}$ is voxel-downsampled and indexed using a kD-tree. Rays are cast from $\mathbf{v}$ to each point and grouped into azimuth–elevation bins (2° resolution). For each bin, only the closest hit is retained, simulating self-occlusion from a specific viewpoint.

(2) **Multi-Object Occlusion**

We place five synthetic spherical occluders with random radii and positions relative to the object centroid. Points falling behind any occluder, based on its radius and viewing direction, are removed to simulate foreground clutter or overlapping vegetation.

(3) **Clustered Random Sampling**

To model irregular dropout, we randomly select seed points and expand a radius-r neighborhood using a kD-tree. These clusters are iteratively removed until a predefined occlusion ratio is reached, mimicking scan noise and partial occlusion due to sensor sparsity.

All occlusion routines are implemented in Python using Open3D Zhou et al. [41] and applied in randomized combinations for each instance to generate diverse and challenging partial scans. The resulting occluded point cloud $\mathscr{P}_{\text{occl}}$ is used to render binary masks S_m for training and evaluation.

2.6. Model architecture

Our amodal segmentation model is trained as a generator in an adversarial setup, and introduces three core architectural contributions:

1. **Coarse-to-Fine Hierarchy:** Contrary to single-scale Vision Transformers (ViTs), our model first processes tokens at the coarsest feature level to capture global shape context. This information is then hierarchically propagated to finer-scale tokens via cross-attention.
2. **Context-Aware Feature Fusion:** The decoder's skip-connections are sourced from the *processed*, context-enriched tokens at every scale of the hierarchy, not the raw encoder features. This design enables each decoder stage to leverage semantically enriched features, thereby enhancing boundary precision and occluded-region recovery.

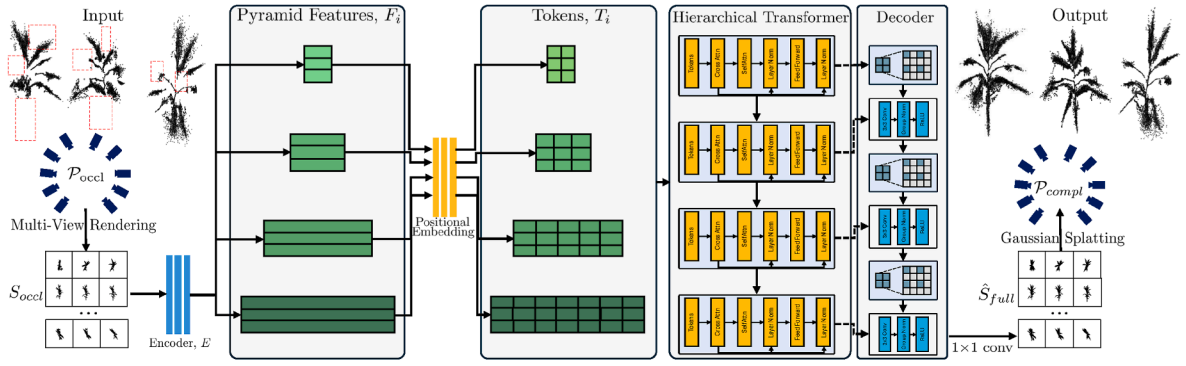

Fig. 1. *A-Occ-Plant* inference overview: We render an occluded point cloud $\mathscr{P}_{occl}$ to get images S_{occl} by positioning the camera on a regularly tessellated sphere with camera intrinsics $\mathbf{K}_{\text{int}}$ and extrinsics $(\mathbf{R}_m, \mathbf{t}_m)$. We feed the images to the Encoder E to extract features F_i at different scales. We tokenize the features with a positional embedding. We feed the tokens into the Hierarchical Transformer and use the decoder. We use a 1×1 convolution layer to refine generated amodal masks $\widehat{S}_{full}$. Then, we use the same $\mathbf{K}_{\text{int}}$ and $(\mathbf{R}_m, \mathbf{t}_m)$ to run Gaussian Splatting without Colmap. The output is a completed 3D point cloud $\mathscr{P}_{compl}$.

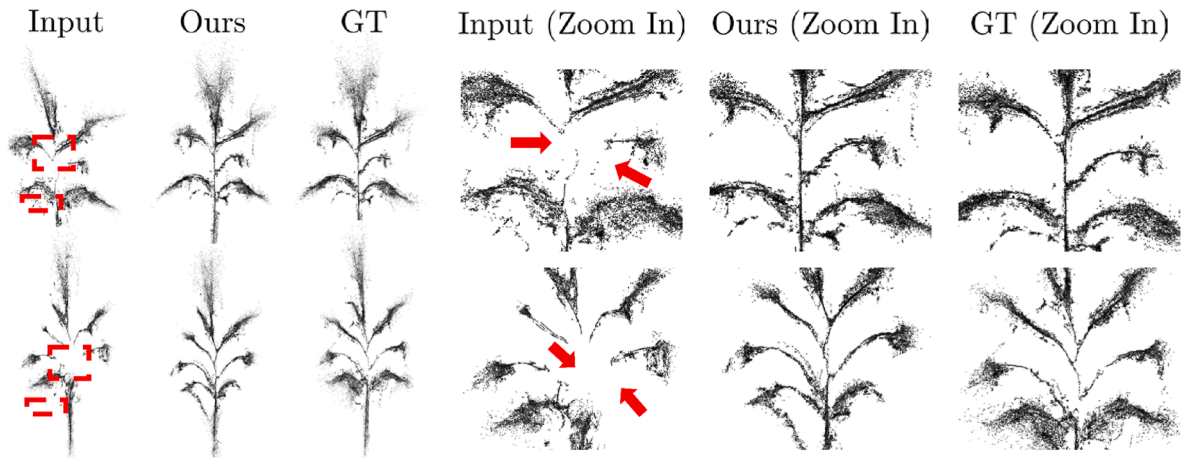

Fig. 2. Detailed view of the completion of occluded point clouds, compared to ground truth (GT) from the maize dataset.

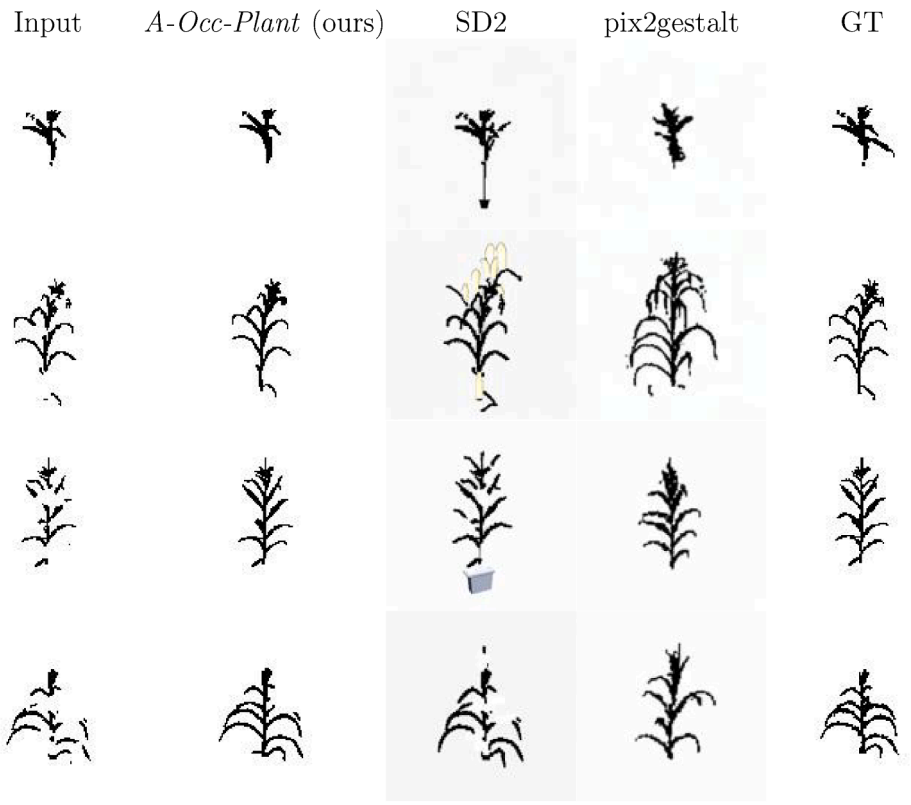

Fig. 3. The visual comparisons from a given input, which is a randomly selected, ours, SD2 Inpainting (SD2), pix2gestalt, and its ground truth (GT) from test images in the maize dataset.

3. **Memory-Efficient and Scalable Design:** Through the integration of memory-efficient attention kernels from the `xFormers` library, the model scales effectively to deep hierarchies without incurring prohibitive computational costs.

2.6.1. *Feature pyramid encoder E*

Given a binary input mask $S \in \{0,1\}^{B\times 1\times H\times W}$, a four-stage convolutional encoder generates a feature pyramid $\{F_i\}_{i=0}^{3}$, where B, H, and W denote a batch size, height, and width dimension, respectively. Each stage i produces features $F_i \in \mathbb{R}^{B\times C_i\times H_i\times W_i}$, where $C_i = 64 \cdot 2^i$. Each stage includes two 3×3 convolutional blocks with GroupNorm (8 groups) and ReLU. Downsampling between stages is performed by a 4×4 transposed convolution with a stride of two.

2.6.2. *Tokenization and positional embedding*

Each feature map F_i is projected into a sequence of $L_i = H_i W_i$ tokens via a 1×1 convolution, yielding an initial token map in $\mathbb{R}^{B\times D\times H_i\times W_i}$, with the token dimension $D = 256$. A learnable 2D positional embedding is added before the spatial dimensions are flattened, resulting in a token sequence $T_i \in \mathbb{R}^{L_i\times B\times D}$. We also create a key-padding mask from the input for use in the attention mechanism, allowing the model to ignore empty regions.

2.6.3. Hierarchical transformer processing

We process tokens in four stages, indexing $i = 3$ (coarsest) to $i = 0$ (finest). A series of transformer blocks governs the process, each applying post-layer normalization.

1. **Cross-Attention:** For the stage $i < 3$, the block integrates context from the coarser level P_{i+1}. Queries are projected from the current level's tokens T_i, while keys and values are projected from the concatenation $[T_i\,;\,P_{i+1}]$.
2. **Self-Attention:** Standard multi-head self-attention (eight heads) refines representations within the current level.
3. **Feed-Forward Network (FFN):** A position-wise MLP with a $4\times$ expansion factor (e.g., $D \to 4D \to D$) is applied.

2.6.4. Cross-scale fusion decoder

The decoder reconstructs the completed mask from the processed token sequences $\{P_i\}_{i=0}^{3}$. Starting with the coarsest features P_3, the decoder iteratively up samples and fuses information from finer levels. Upsampling is performed by a 4×4 transposed convolution with a stride of two. The upsampled features are concatenated channel-wise with the corresponding processed tokens P_i from the hierarchy. This is followed by a 3×3 convolutional block with GroupNorm and ReLU. A final 1×1 convolution projects the feature map to a single-channel logit map, predicting the soft amodal mask $\widehat{S}_{full} \in \mathbb{R}^{B\times 1\times H\times W}$.

2.7. Optimization and loss function

Given a partial mask S_{occl} and its ground-truth full mask S_{full}, the model f_θ produces per-pixel logits

$$L_m(u,v) = f_\theta(S_{occl})(u,v), \quad \widehat{S}_{full}(u,v) = \sigma(L_m(u,v)).$$

We apply a per-pixel weight

$$w(u,v) = \begin{cases} w_b, & S_{\text{full}}(u,v) = 0, \\ 1, & S_{\text{full}}(u,v) = 1, \end{cases} \quad w_b = 1.0,$$

and define the weighted reconstruction losses as

$$\mathscr{L} = \lambda_{L1}\mathscr{L}_{\text{L1}} + \mathscr{L}_{\text{BCE}}, \quad \lambda_{L1} = 2.0.$$

We optimize θ with Adam Kingma and Ba [42] optimizer ($\eta = 2 \times 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$) using a batch size of 128. At each step we compute $\nabla_\theta \mathscr{L}$ and update $\theta \leftarrow \theta - \eta\nabla_\theta \mathscr{L}$.

2.8. 3D reconstruction via Gaussian Splatting

We initialize the Gaussian Splatting Kerbl et al. [26] with 100,000 3D Gaussian random primitives and the default hyperparameters. Since the camera parameters are known at the rendering stage, our method does not require an external structure-from-motion tool, such as COLMAP Schönberger and Frahm [43]. This significantly reduces runtime.

For each view m and pixel (u, v) with soft confidence $\widehat{S}_{full}(u,v)$, we compute $\mathbf{d}_{m,u,v}$, which is the unit ray from view m to pixel (u, v) in world coordinates and place an isotropic Gaussian $\mathcal{N}\big(\mathbf{x};\, r\,\mathbf{d}_{m,u,v},\, \sigma^2\mathbf{I}\big)$ at the fixed depth r. Each Gaussian is weighted by $\widehat{S}_{full}(u,v)$ and accumulated into the continuous density field

$$\rho(\mathbf{x}) = \sum_{m=1}^{M}\sum_{u=1}^{W}\sum_{v=1}^{H} \widehat{S}_{full}(u,v)\,\mathcal{N}\big(\mathbf{x};\, r\,\mathbf{d}_{m,u,v},\, \sigma^2\mathbf{I}\big).$$

The final point cloud $\widehat{\mathscr{P}}$ is recovered by sampling the highest-density locations in $\rho(\mathbf{x})$, yielding seamless, high-fidelity reconstructions as described in Gaussian Splatting.

3. Experiments

3.1. Implementation details

The system was implemented in Python using PyTorch v2.7.1+cu126 and PyTorch Lightning. We used an NVIDIA H100 GPU with 80 GB of RAM and trained for 100 epochs with a batch size of 128 in mixed-precision FP16 mode.

We validate our training model using quantitative and qualitative analysis. We use two strong baselines: the current publicly available SOTA, pix2gestalt Ozguroglu et al. [13], and a stable-diffusion-2-based inpainting model. We strictly follow their default settings for the training on our datasets. For the inpainting model, we designed a detailed text prompt: `A single completed (maize | sorghum) plant without any occlusions, great detail on a plain white background.` to obtain $\widehat{S}_{full}$. We also show ablation studies.

3.2. Datasets

We use two datasets: one comprising 1046 scanned maize plants Kimara et al. [40] and the other comprising 1000 realistic synthetic sorghum 3D point clouds from a procedural model Ostermann et al. [2]. All code used for the dataset generation will be released. We note that while the maize clouds are real field scans and the sorghum clouds are high-fidelity procedural models, in both cases the occluded inputs are not naturally collected partial scans: they are produced from the complete point clouds with the synthetic occlusion generator described in Sec. 2.5. This yields paired (occluded, complete) supervision and exact ground truth for quantitative evaluation, at the cost of not modeling every artifact of a strictly real partial scan.

3.2.1. Maize dataset

We evaluate our approach on a field-scanned maize dataset Kimara et al. [40] containing 1046 plants, each represented by 10,000 points. We randomly split the dataset into 836 training and 209 testing plants (80/20 split). For each plant, we render paired full S_{occl} and occluded binary masks S_{full}, generating 200 views per instance. This yields 167, 200 training and 42,000 test images. Applying our synthetic occlusion generator produces an average occlusion rate of 43.1% across all views.

3.2.2. Sorghum dataset

For a controlled synthetic benchmark, we generate 1000 sorghum plant models using a procedural framework Ostermann et al. [2]. Each instance is represented as 16,384 points. Adopting an 80/20 train/test split, we render paired full S_{full} and occluded masks S_{occl}, yielding 200 views per plant with 160,000 training and 40,000 test images in total. The resulting dataset contains an average occlusion rate of 39.6%.

3.3. Evaluation metrics

We evaluate completed masks and reconstructions using five metrics:

- **Intersection-over-Union (IoU)(↑).** For each predicted soft mask $\widehat{S}_{full}$, we binarize at 0.5 and compute which ranges from zero (no overlap) to one (perfect overlap).
- **Structural Similarity Index Measure (SSIM)(↑)** assesses perceptual similarity between the predicted mask $\widehat{S}_{full}$ and ground truth S_{full}.
- **Learned Perceptual Image Patch Similarity (LPIPS) (↓)** Zhang et al. [44] measures deep perceptual distance using pretrained VGG features.
- **Chamfer Distance (CD) (↓).** Given the predicted point set $\widehat{\mathscr{P}}$ and ground truth $\mathscr{P}$, we compute the bidirectional Chamfer distance.

Table 1
We validate amodal segmentation quality on the maize and sorghum test sets, comparing *A-Occ-Plant* against pix2gestalt and SD2 Inpainting using four metrics (LPIPS, SSIM, IoU, PSNR). We also report the number of trainable parameters and the number of images generated per second; these are model properties and therefore identical across datasets. The best value per metric within each dataset is in **bold**.

Method	Maize				Sorghum				Parameters (↓)	image/sec (↑)
	LPIPS (↓)	SSIM (↑)	IoU (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	IoU (↑)	PSNR (↑)		
pix2gestalt	0.197	0.841	0.954	13.403	0.232	0.742	0.925	11.382	860 M	0.489
SD2 Inpainting	0.126	0.876	0.968	14.968	0.115	0.858	0.963	14.540	-	2.059
A-Occ-Plant (ours)	**0.054**	**0.917**	**0.978**	**16.761**	**0.041**	**0.924**	**0.984**	**18.169**	**20 M**	**19.291**

- **F1 Score @1%. (↑)** We set the distance threshold $t = 0.01 \times d_{\text{diag}}$, where d_{diag} is the diagonal of the ground-truth bounding box. Precision and recall are computed by counting points within t.

3.4. Quantitative analysis

3.4.1. 2D amodal segmentation

Table 1 shows the comparison of *A-Occ-Plant* against pix2gestalt Ozguroglu et al. [13], trained on the same maize dataset using its default configuration. Our approach reduces LPIPS by a factor of 2.65 and achieves relative improvements of 8.3% in SSIM, 2.5% in IoU, and 20.0% in PSNR. Remarkably, our model uses only 2.3% of the parameters of pix2gestalt and achieves an inference speed of 19 images per second, compared to just 0.5 images per second for pix2gestalt, yielding a 39.4× speedup.

The most significant improvement is observed in LPIPS, indicating that our hierarchical Transformer generates perceptually more faithful completions. The model produces coherent boundaries and fills occluded regions with semantically plausible content by propagating global shape information from coarse to fine scales through cross-attention.

Using the sorghum dataset (see Table 1) results in a 19.7% relative gain in SSIM compared to pix2gestalt, demonstrating improved structural fidelity, which is further supported by context-aware feature fusion in the decoder. The 6.0% increase in IoU reflects more accurate mask overlap, aided by global context that guides the recovery of occluded regions. A 37.3% relative improvement in PSNR suggests reduced pixel-level reconstruction error, driven by the model's ability to jointly preserve sharp boundaries and smooth interiors. These results indicate that our architecture effectively combines global priors with fine-grained details, outperforming the state of the art across all key metrics.

3.4.2. 3D point cloud

From the completed amodal masks $\widehat{S}_{full}$, we leverage Gaussian splatting using the original camera intrinsics $\mathbf{K}$ and extrinsics ($\mathbf{R}_m$, $\mathbf{t}_m$), thus eliminating the need for external pose estimation. To assess reconstruction quality, we report the bidirectional Chamfer Distance (CD) and F1@1%. Both metrics are standard for evaluating point-cloud fidelity.

Table 2 presents results on the maize dataset, comparing our method against Pix2Gestalt Ozguroglu et al. [13] and a Stable Diffusion-based inpainting baseline. Table 3 shows a comparison with point cloud completion methods. Our approach achieves a 176% and 178% relative reduction in CD, and a 154% and 56% relative increase in F1@1%, respectively. On the sorghum dataset (Table 2), we observe 163% and 8% relative improvements in CD, and 580% and 64% relative gains in F1@1%.

These experiments indicate that our coarse-to-fine hierarchical Transformer and cross-scale fusion decoder generate more accurate and consistent 3D reconstructions than the state-of-the-art 2D amodal segmentations.

Table 2
Quantitative evaluation of 3D reconstruction quality on the maize and sorghum datasets using Gaussian splatting with our predicted masks $\widehat{S}_{full}$. We report Chamfer Distance and F1@1% metrics. The best result per metric within each dataset is in **bold**.

Method	Maize		Sorghum	
	Chamfer Distance (↓)	F1@1% (↑)	Chamfer Distance (↓)	F1@1% (↑)
pix2gestalt	1.049	0.145	1.994	0.076
SD2 Inpainting	1.058	0.237	0.816	0.314
A-Occ-Plant (ours)	**0.380**	**0.369**	**0.758**	**0.515**

3.5. Qualitative analysis

3.5.1. 2D amodal segmentation

We validate the quality of amodal segmentation in the test split from the Maize and Sorghum dataset. Fig. 3 shows the task from a given input (first column) to make it as accurate as the last column (its ground truth). Our approach completes the missing pixel information using diverse viewpoints. In the third column, the stable-diffusion-2 inpainting Rombach et al. [45] model failed to complete the image and insert objects into the scene with a non-white background. The outputs in the fourth column come from the SOTA Ozguroglu et al. [13]. They complete the images but fail to preserve the overall appearance of the given input. See more results in the appendix.

This trend is also visible in the Sorghum dataset (Fig. 4). In the second column, ours generates complete images in detail even in high-frequency areas, e.g. leaves around the stem (second row). Yet, stable-diffusion-2 inserts a random object or fails to complete the missing pixels (fourth row). All missing pixels are completed using pix2gestalt, but the alignment (first and third rows) is a main issue, and the unmatched scale (second and fourth rows) is also noted as a limitation.

3.5.2. 3D point cloud

Figs. 5 and 6 present qualitative 3D reconstructions of the maize and the sorghum dataset, comparing our method against two baselines: Stable Diffusion 2-based inpainting (SD2) and Pix2Gestalt. From left to right, each column shows the partial scan, the ground truth, and the reconstructions generated by SD2, Pix2Gestalt, and our approach using Gaussian splatting of the completed amodal masks $\widehat{S}_{full}$.

Our method generates point clouds that capture stems and leaves, which are missing or fragmented in the partial inputs. In contrast, SD2 often introduces noise and isolated points, resulting in cluttered 3D geometry. Pix2Gestalt's in painted masks suffer from local inconsistencies, which introduce holes and distortions in the generated point clouds. By leveraging a coarse-to-fine hierarchical Transformer for 2D amodal completion and an efficient multi-view fusion, our pipeline produces clean, high-fidelity 3D models that closely match the ground truth, as we show in Fig. 2.

Table 3
Downstream phenotypic trait recovery on maize (209-plant test set) and soybean (21 held-out plants): mean absolute error (MAE) and Pearson correlation against ground truth. Leaf count is the number of stem-suppressed DBSCAN clusters, leaf angle is the mean per-leaf PCA elevation, and plant height is the vertical extent. All clouds are unit-sphere normalized, so traits are pose- and scale-invariant. Best per row within each crop in **bold**.

Trait	Maize				Soybean			
	PointAttN	AdaPoinTr	AnchorFormer	*A-Occ-Plant* (ours)	PointAttN	AdaPoinTr	AnchorFormer	*A-Occ-Plant* (ours)
Leaf count (MAE ↓)	4.55	4.55	5.05	**2.21**	12.7	12.4	13.1	**7.2**
Leaf count (corr ↑)	0.25	0.26	0.04	**0.34**	0.19	0.29	0.16	**0.51**
Leaf angle° (MAE ↓)	26.9	29.6	31.7	**10.8**	13.7	10.4	13.4	**7.4**
Leaf angle (corr ↑)	0.27	0.23	0.23	**0.45**	–	–	–	–
Plant height (MAE ↓)	0.17	**0.16**	0.21	0.16	0.172	0.147	0.182	**0.126**
Plant height (corr ↑)	0.36	0.35	−0.06	**0.44**	0.43	0.56	0.36	**0.69**

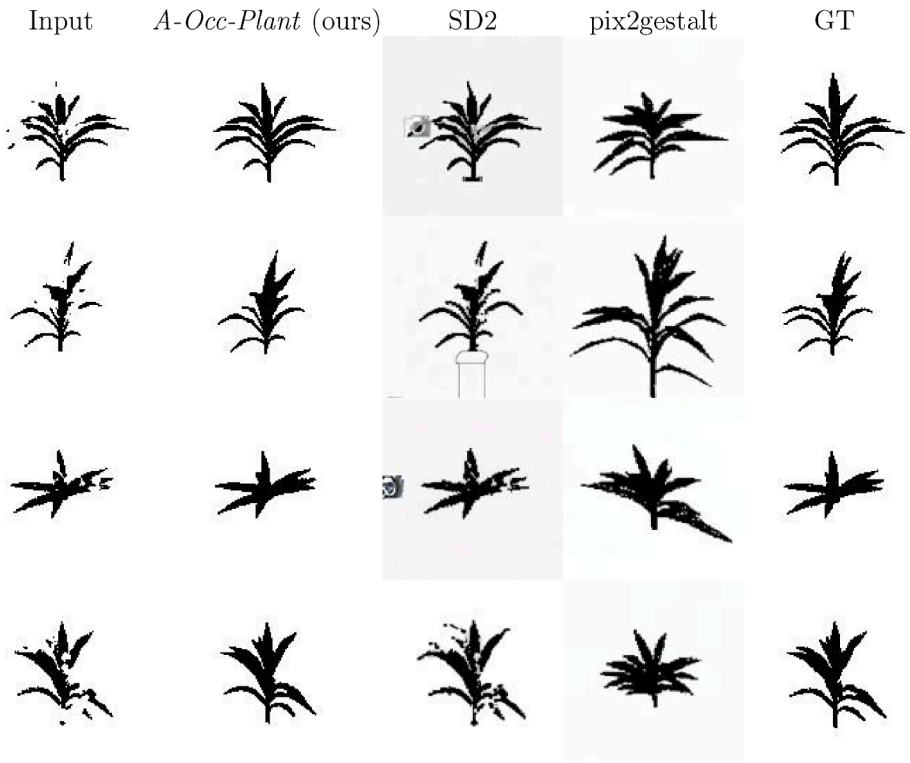

Fig. 4. The visual comparisons from a given input, which is a randomly selected one, to ours, SD2 Inpainting (SD2), pix2gestalt, and its ground truth (GT) from test images in the sorghum dataset.

3.6. Comparison with 3D point-cloud completion methods

We trained three 3D point cloud completion methods Yu et al. [28]; Chen et al. [29]; Wang et al. [27] using the same dataset. Fig. 7 shows that their clouds are noisy and lose the leaf-and-stem structure, whereas *A-Occ-Plant* closely matches the ground truth. (see Fig. 1).

3.7. Generalization to soybean

We show that our work can be transferred to a crop with a different morphology by evaluating on the Soybean-MVS dataset Sun et al. [39], which provides 102 real soybean plants. We adopt an 81/21 train/test split. Fig. 8 shows that ours preserves a structure from the occluded input soybean to get the ground truth data, but other baselines over-predict surface area.

3.8. Downstream phenotypic trait recovery

LPIPS, SSIM, IoU, PSNR, and Chamfer Distance measure reconstruction quality, not phenotyping utility. We demonstrate phenotyping capability by extracting three biologically meaningful traits from each reconstruction using the Maize dataset: leaf count (stem-suppressed DBSCAN clusters), mean leaf angle (per-leaf PCA elevation), and plant height (vertical extent). Table 3 shows that ours yields 2× lower leaf-count error (2.21 vs. 4.55-5.05) and 3× lower leaf-angle error (10.8° vs. 26.9°-31.7°). This trend consistently holds using the soybean Sun et al. [39] dataset. Our method achieves the lowest error and the highest correlation across all traits. In particular, ours shows less than 42% error compared to the strongest baseline, AdaPoinTr, and the leaf-count correlation from ours is 76% higher. And the leaf-angle MAE is only 7.40°, which is 45% more accurate than AnchorFormer. Overall, *A-Occ-Plant* improves every phenotypic trait by 14-76% compared to the best competing method.

3.9. Ablation study

We quantify the impact of key design choices on the maize test set in Table 4, with our default configuration (64 ch, 8 heads, 4 stages, 256-dim tokens, learned embeddings) highlighted.

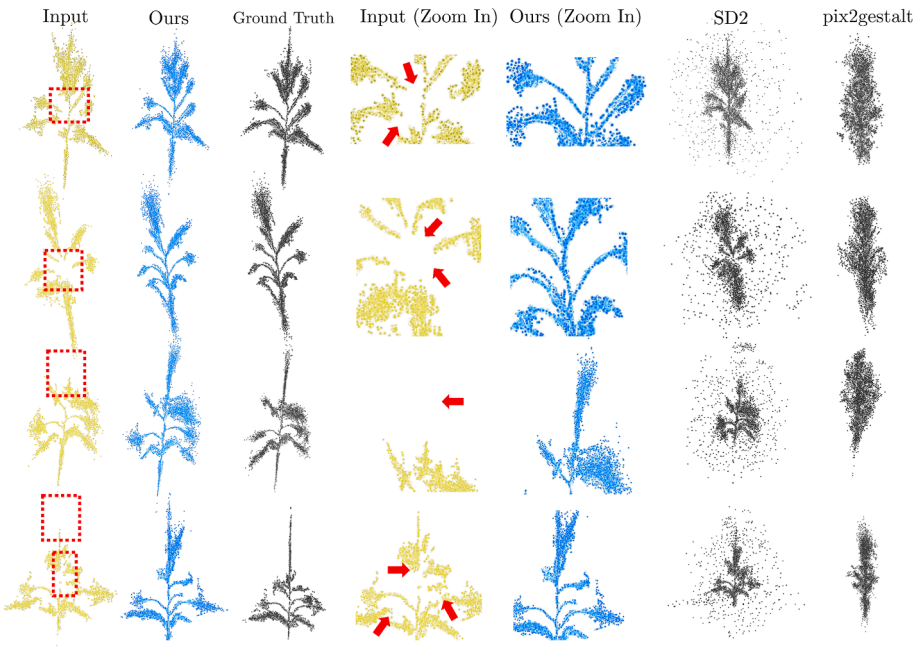

Fig. 5. Qualitative 3D reconstruction results on the maize dataset. From left to right: partial scan, ground truth, highlighted missing region, close-up views, and reconstructions from SD2 and pix2gestalt.

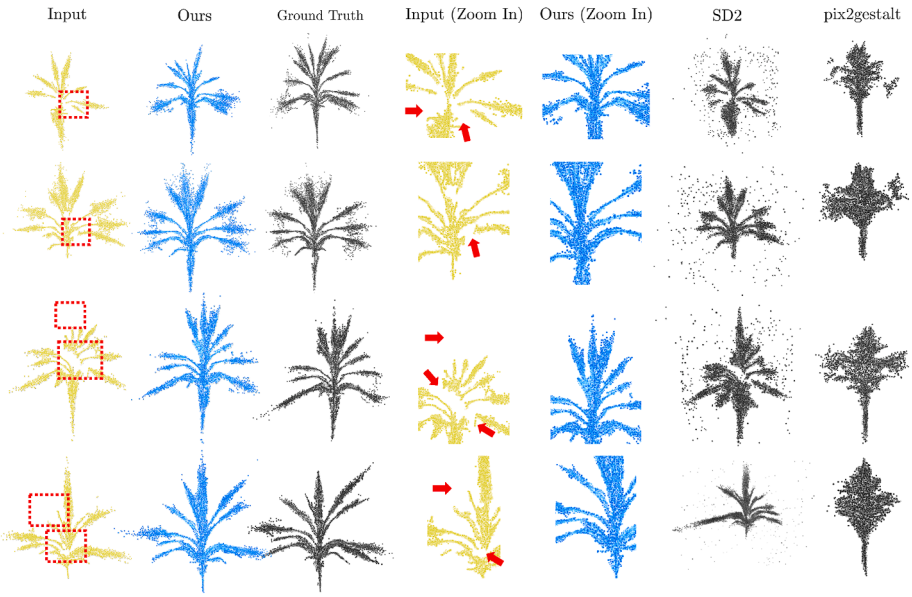

Fig. 6. Qualitative 3D reconstruction results on the sorghum dataset. From left to right: partial scan, ground truth, highlighted missing region, close-up views, and reconstructions from SD2 and pix2gestalt.

Token dimension Halving from 256 to 128 raises LPIPS by 7%, and reduces SSIM/IoU by 1% each, with negligible PSNR loss—indicating the need for a rich token space.

Attention heads Reducing to 4 heads degrades LPIPS by 8%, SSIM by 1%, IoU by 2%, and PSNR by 3%. A single head matches LPIPS but incurs an additional 2% PSNR drop, indicating that multi-head attention is beneficial.

Positional embeddings Removing learned 2D embeddings increases LPIPS by 49% and lowers SSIM/IoU by 0.3%/2%, confirming that spatial encoding is critical.

Number of stages Cutting to 2 stages worsens LPIPS by 11%, SSIM/IoU by 1%, and PSNR by 3%. A single-stage model collapses performance (LPIPS +56%, SSIM/IoU −6%, PSNR –6%), emphasizing the value of coarse-to-fine hierarchy.

Channel width Halving channels to 32/16 increases LPIPS by 41%/49% and drops SSIM/PSNR by about 1%, demonstrating the need for sufficient feature capacity.

Number of views As shown in Fig. 9, performance plateaus beyond 200 rendered views (220/230 show less than 1% gains), so we adopt 200 views.

Overall, the default setting achieves the best balance across all metrics, validating our architectural choices.

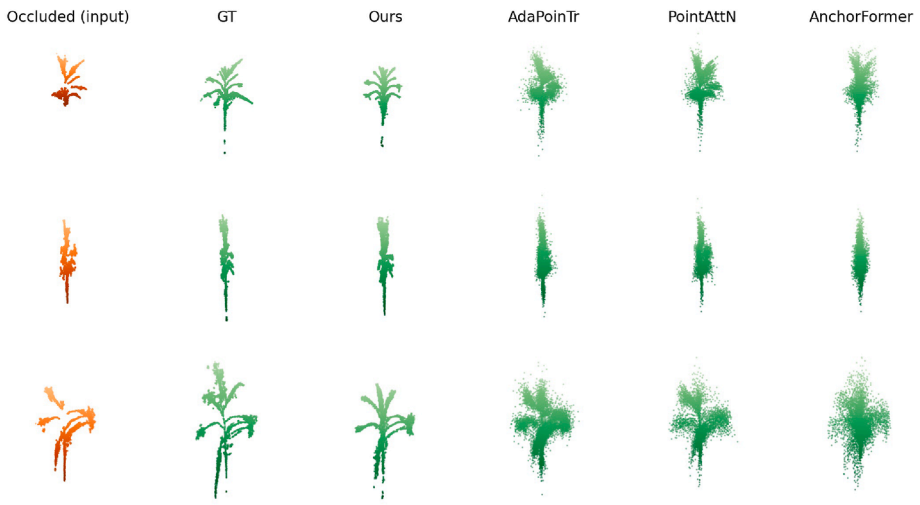

Fig. 7. Qualitative maize reconstructions using 3D point cloud completion methods (AdaPoinTr Yu et al. [28], PointAttN Wang et al. [27], AnchorFormer Chen et al. [29]). From left to right: the occluded input, the ground-truth complete cloud, our reconstruction, and baselines. Our method preserves the plant structure while baselines do not.

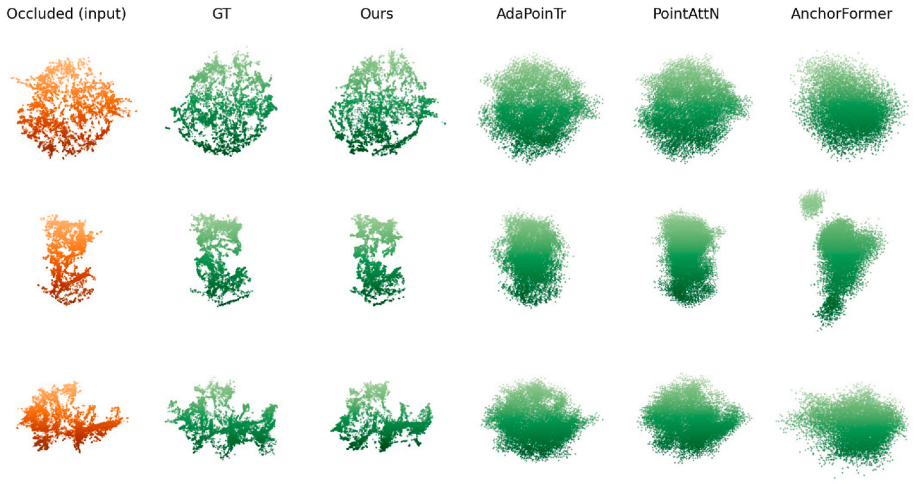

Fig. 8. Qualitative soybean reconstructions using the soybean dataset. From left to right: the occluded input, the ground-truth complete cloud, our reconstruction, and the three 3D point-cloud completion baselines (AdaPoinTr, PointAttN, AnchorFormer). Our algorithm preserves the plant structure, whereas other baselines tend to overpredict surface regions, producing overly bushy shapes.

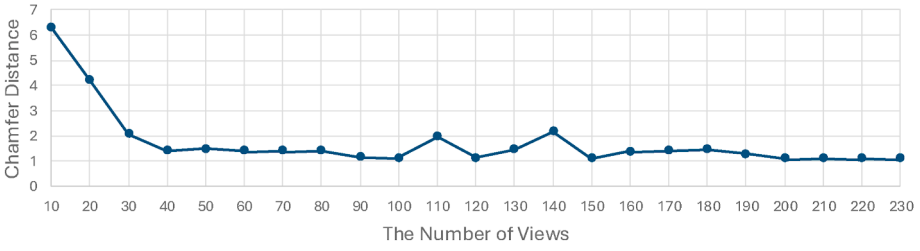

Fig. 9. Effect of the number of input views on 3D reconstruction quality, measured by Chamfer Distance.

4. Discussion

4.1. Limitations

Our method comes from the intrinsic differences in sampling between the point cloud density and the mask resolution. We are limited by a low resolution, which maps many points to a single pixel, and increasing the resolution will only push the problem to a higher frequency. While this is a limitation, the method has its niche in point cloud completion for small-scale plants in enclosed environments, such as phenotyping facilities. Future work should thus focus on large plants.

Another limitation is coming from the syn2real gap. The point clouds are real (field-scanned maize, multi-view-stereo soybean) or high-fidelity procedural (sorghum), but the occluded-complete pairs used for supervision and evaluation are not naturally collected partial scans: they are constructed by applying a controlled synthetic occlusion

Table 4
Ablation of the impact of different parameter settings on all metrics. The highlighted first row is our default.

Channels	Heads	Stages	Token Dim.	Emb.	LPIPS (↓)	SSIM (↑)	IoU (↑)	PSNR (↑)
64	8	4	256	✓	**0.054**	**0.917**	**0.978**	**16.761**
64	8	4	*128*	✓	0.058	0.912	0.969	16.721
64	*4*	4	256	✓	0.059	0.910	0.962	16.341
64	*1*	4	256	✓	0.059	0.912	0.963	16.407
64	8	4	256	✗	0.106	0.914	0.962	16.426
64	8	*2*	256	✓	0.061	0.907	0.969	16.325
64	8	*1*	256	✓	0.123	0.864	0.920	15.825
32	8	4	256	✓	0.092	0.925	0.973	16.522
16	8	4	256	✓	0.107	0.913	0.961	16.422

process to the complete point clouds. This design is deliberate and provides an exact paired ground truth, enabling the controlled, quantitative evaluation reported above. While the syn2real procedure is commonly used, it does not reproduce all artifacts of a strictly real partial scan (e.g., sensor noise, registration errors, or material-dependent dropout). Bridging the gap can be achieved by improving the synthetic component of the process or by restoring real partial scans.

4.2. Future work

Future work should focus on scaling the approach to **large plants**, which will likely require a hierarchical or patch-based processing pipeline rather than the current unit-sphere normalization.

A key area for improvement is addressing the **mismatch between 3D point density and 2D mask resolution**. This could be explored by investigating adaptive rendering techniques, in which mask resolution is dynamically adjusted, or by developing hybrid models that operate directly on dense 3D point clouds while using 2D segmentation for broader context.

Finally, while our hierarchical Transformer shows strong performance, the pipeline's 2D completion module is swappable. Future work could **integrate next-generation 2D diffusion priors** or foundation models as the amodal segmentation, which may offer enhanced generative capabilities and even stronger multi-view consistency.

5. Conclusion

We introduced a novel pipeline for completing partial 3D plant scans by combining multi-view 2D amodal segmentation with Gaussian splatting. Our method employs dense rendering, a hierarchical Transformer for coarse-to-fine mask completion, and an efficient fusion strategy that avoids external pose estimation and fixed-size inputs. Validation on real and synthetic datasets shows that our approach outperforms state-of-the-art baselines by a large margin, achieving up to 264.8% LPIPS improvement while being 39.4× faster with only 2.3% of the parameters.

Authors' contributions

J. Lee and B. Benes designed the project. J. Lee implemented and conducted the experiments. J. Lee and B. Benes wrote the manuscript, supervised the research, and provided guidance. All authors reviewed and approved the final manuscript.

Funding

The GPU resource is supported by NSF NAIRR Pilot (NAIRR250105). This work was supported in part by the U.S. National Science Foundation under awards No. 2412928and 2417510. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

This work is based upon efforts supported by the USDA-NIFA grants 2024-67021-42879 "Improving Forest Management Through Automated Measurement of Tree Geometry" and 2023-08563 "Optimizing soybean canopy architecture for efficient light and water use under climate change". The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government or NRCS. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank the shared Maize Kimara et al. [40] and Sorghum Ostermann et al. [2] Dataset.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.plaphe.2026.100240.

Data availability

The inference code and sample data used in this study are available. The project can be downloaded from https://drive.google.com/file/d/1UO7V7fdOsLuI9qemCODxoc50WZYsuuv3/view?usp=sharing. After downloading and unzipping the file, please refer to the included README.md for detailed instructions on reproducing the results. The full code and data at https://github.com/JaeLee18/PlantPhenomics_Occlusion.

References

[1] Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, Kui Jia, Surface reconstruction from point clouds: a survey and a benchmark, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[2] Ian Ostermann, Bedrich Benes, Mathieu Gaillard, Bosheng Li, Jensina Davis, Ryleigh Grove, Nikee Shrestha, Michael C. Tross, James C. Schnable, Sorghum segmentation and leaf counting using in silico trained deep neural model, Plant Phenom. J. 7 (1) (2024) e70002, https://doi.org/10.1002/ppj2.70002.
[3] Jae Joong Lee, Bosheng Li, Sara Beery, Jonathan Huang, Songlin Fei, Raymond A. Yeh, Bedrich Benes, Tree-d fusion: simulation-ready tree dataset from single

images with diffusion priors, in: European Conference on Computer Vision, Springer, 2024, pp. 439–460.
[4] Jorge Vazquez, Shuwen Yang, Yiqun Zhang, Jeffrey Demieville, Brennan Huppenthal, Nirav Merchant, Voicu Popescu, Alejandra Magana, Duke Pauli, Bedrich Benes, Vr biotalk—Hands-free visual analytics of phenotyping data using natural conversation, IEEE Access 14 (2026) 75577–75592, https://doi.org/10.1109/ACCESS.2026.3692426.
[5] Wanquan Yang, Hu Feng, Xinyu Zhang, Yatao Zhang, Joel Banh, Felicia Viselli, et al., Crop phenomics and high-throughput phenotyping: technology and trends, Front. Plant Sci. 11 (2020) 1–32.
[6] Yassine Himeur, Bhagawat Rimal, Abhishek Tiwary, Abbes Amira, Using artificial intelligence and data fusion for environmental monitoring: a review and future perspectives, Inf. Fusion 86 (2022) 44–75.
[7] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, Martial Hebert, Pcn: point completion network, in: 3D Vision (3DV), 2018 International Conference on, 2018.
[8] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, Jie Zhou, Pointr: diverse point cloud completion with geometry-aware transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12498–12507.
[9] Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, Scpnet: semantic scene completion on point cloud, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 17642–17651.
[10] Yi Rong, Haoran Zhou, Lixin Yuan, Cheng Mei, Jiahao Wang, Lu Tong, Cra-pcn: point cloud completion with intra-and inter-level cross-resolution transformers, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
[11] Minh Tran, Khoa Vo, Kashu Yamazaki, Arthur Fernandes, Michael Kidd, Ngan Le Aisformer, Amodal Instance Segmentation with Transformer, BMVC, 2022.
[12] Jianxiong Gao, Xuelin Qian, Yikai Wang, Tianjun Xiao, Tong He, Zheng Zhang, Yanwei Fu, Coarse-to-fine amodal segmentation with shape prior, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1262–1271.
[13] Ege Ozguroglu, Ruoshi Liu, Dídac Surś, Dian Chen, Achal Dave, Pavel Tokmakov, Carl Vondrick, pix2gestalt: Amodal Segmentation by Synthesizing Wholes, CVPR, 2024.
[14] Guanqi Zhan, Chuanxia Zheng, Weidi Xie, Andrew Zisserman, Amodal ground truth and completion in the wild, in: CVPR, 2024.
[15] Katherine Xu, Lingzhi Zhang, Jianbo Shi, Amodal completion via progressive mixed context diffusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 9099–9109.
[16] Jae Joong Lee, Bedrich Benes, Raymond A. Yeh, Tuning-free amodal segmentation via the occlusion-free bias of inpainting models, in: Proceedings of the AAAI Conference on Artificial Intelligence, 40, 2026, pp. 5872–5880, https://doi.org/10.1609/aaai.v40i7.37509.
[17] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, Silvio Savarese, Topnet: structural point cloud decoder, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 383–392.
[18] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, Wenxiu Sun, Grnet: gridding residual network for dense point cloud completion, in: ECCV, 2020.
[19] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, Zhizhong Han, Snowflakenet: point cloud completion by snowflake point deconvolution with skip-transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5499–5509.
[20] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Zheng Wen, Yu-Shen Liu, Pmp-net++: point cloud completion by transformer-enhanced multi-step point moving paths, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2022) 852–867.
[21] Pingping Cai, Canyu Zhang, Lingjia Shi, Lili Wang, Nasrin Imanpour, Song Wang, Einet: point cloud completion via extrapolation and interpolation, in: European Conference on Computer Vision, Springer, 2024, pp. 377–393.
[22] Tianxin Huang, Zhiwen Yan, Yuyang Zhao, Gim Hee Lee, Compc: completing a 3d point cloud with 2d diffusion priors, arXiv preprint arXiv:2404.06814 (2024).
[23] Xiaochen Zhou, Bosheng Li, Bedrich Benes, Ayman Habib, Songlin Fei, Jinyuan Shao, Soeren Pirk, Trees tructor: forest reconstruction with neural ranking, IEEE Trans. Geosci. Rem. Sens. 63 (2025) 1–19, https://doi.org/10.1109/TGRS.2025.3558312.
[24] Ariyan Zarei, Bosheng Li, James C. Schnable, Eric Lyons, Duke Pauli, Kobus Barnard, Bedrich Benes Plantsegnet, 3d point cloud instance segmentation of nearby plant organs with identical semantics, Comput. Electron. Agric. 221 (2024) 108922, https://doi.org/10.1016/j.compag.2024.108922. ISSN 0168-1699.
[25] Yanchao Liu, Jianwei Guo, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, Hui Huang, Treepartnet: neural decomposition of point clouds for 3d tree reconstruction, ACM Trans. Graph. 40 (6) (December 2021) 1–16, https://doi.org/10.1145/3478513.3480486.
[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, 3d gaussian splatting for real-time radiance field rendering, ACM Trans. Graph. 42 (4) (July 2023). URL, https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.
[27] Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, Chunhua Shen, Pointattn: you only need attention for point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 5472–5480.
[28] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, Jie Zhou, Adapointr: diverse point cloud completion with adaptive geometry-aware transformers, URL, https://arxiv.org/abs/2301.04545, 2023.
[29] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, Tao Mei, Anchorformer: point cloud completion from discriminative nodes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13581–13590.
[30] Yan Zhu, Yuandong Tian, Dimitris Metaxas, Piotr Dollár, Semantic amodal segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1464–1472.
[31] Kiana Ehsani, Roozbeh Mottaghi, Ali Farhadi, Segan: segmenting and generating the invisible, in: CVPR, 2018.
[32] Ziheng Zhang, Anpei Chen, Ling Xie, Jingyi Yu, Shenghua Gao, Learning semantics-aware distance map with semantics layering network for amodal instance segmentation, in: ACM MM, 2019.
[33] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, Chen Change Loy, Self-supervised scene de-occlusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3784–3792.
[34] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, Shenghua Gao, Amodal segmentation based on visible region segmentation and shape prior, in: AAAI, 2021.
[35] Yan Zhu, Yuandong Tian, Dimitris Metaxas, Piotr Dollár, Semantic amodal segmentation, in: CVPR, 2017.
[36] Yan Zhu, Yuandong Tian, Dimitris Metaxas, Piotr Dollár, Semantic amodal segmentation, in: CVPR, 2017.
[37] Zhaochen Liu, Limeng Qiao, Xiangxiang Chu, Lin Ma, Tingting Jiang, Towards efficient foundation model for zero-shot amodal segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 20254–20264.
[38] Kaihua Chen, Deva Ramanan, Tarasha Khurana, Using diffusion priors for video amodal segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 22890–22900.
[39] Yongzhe Sun, Zhixin Zhang, Kai Sun, Shuai Li, Jianglin Yu, Linxiao Miao, Zhanguo Zhang, Yang Li, Hongjie Zhao, Zhenbang Hu, et al., Soybean-mvs: Annotated three-dimensional model dataset of whole growth period soybeans for 3d plant organ segmentation, Agriculture 13 (7) (2023) 1321.
[40] Elvis Kimara, Mozhgan Hadadi, Godbersen Jackson, Aditya Balu, Talukder Jubery, Yawei Li, Adarsh Krishnamurthy, Patrick S. Schnable, Baskar Ganapathysubramanian, Agrifield3d: a curated 3d point cloud and procedural model dataset of field-grown maize from a diversity panel, arXiv preprint arXiv:2503.07813 (2025).
[41] Qian-Yi Zhou, Jaesik Park, Vladlen Koltun, Open3D: a modern library for 3D data processing, arXiv:1801.09847 (2018).
[42] Diederik P. Kingma, Jimmy Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[43] Johannes L. Schönberger, Jan-Michael Frahm, Structure-from-Motion revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[44] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, Oliver Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: CVPR, 2018.
[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 10684–10695.