Complex & Intelligent Systems (2024) 10:3389–3409
https://doi.org/10.1007/s40747-023-01325-8

ORIGINAL ARTICLE

Point cloud completion network for 3D shapes with morphologically diverse structures

Chun-Jing Si[1,5] · Zhi-Ben Yin[2] · Zhen-Qi Fan[1] · Fu-Yong Liu[2] · Rong Niu[3] · Na Yao[1,5] · Shi-Quan Shen[1] · Ming-Deng Shi[1,5] · Ya-Jun Xi[4]

Received: 27 August 2023 / Accepted: 13 December 2023 / Published online: 2 February 2024

Abstract

Point cloud completion is a challenging task that involves predicting missing parts in incomplete 3D shapes. While existing strategies have shown effectiveness on point cloud datasets with regular shapes and continuous surfaces, they struggled to manage the morphologically diverse structures commonly encountered in real-world scenarios. This research proposed a new point cloud completion method, called SegCompletion, to derive complete 3D geometries from a partial shape with different structures and discontinuous surfaces. To achieve this, morphological segmentation was introduced before point cloud completion by deep hierarchical feature learning on point sets, and thus, the complex morphological structure was segmented into regular shapes and continuous surfaces. Additionally, each instance of a point cloud that belonged to the same type of feature could also be effectively identified using HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise). Furthermore, the multiscale generative network achieved sophisticated patching of missing point clouds under the same geometric feature based on feature points. To compensate for the variance in the mean distances between the centers of the patches and their closest neighbors, a simple yet effective uniform loss was utilized. A number of experiments on ShapeNet and Pheno4D datasets have shown the performance of SegCompletion on public datasets. Moreover, the contribution of SegCompletion to our dataset (Cotton3D) was discussed. The experimental results demonstrated that SegCompletion performed better than existing methods reported in the literature.

Keywords Morphologically diverse structures · Morphological segmentation · Uniform loss · Point cloud completion · SegCompletion

Introduction

Point clouds become easier to capture by laser scanners [1], LiDAR [2, 3], RGB-D scanners [4], stereo cameras [5], and so on. They have become the primary data format for representing the 3D world and can preserve the original geometric information of objects. They have garnered significant attention in various research fields, including virtual reality, robotics, autonomous driving, 3D games, and automatic plant phenotyping [6–8]. However, the direct raw point clouds obtained from these devices are predominantly sparse and captured only partially since limitations of occlusions, device resolution, transparency, shooting angles, and reflections. Especially, for complicated morphological objects, characterized by significant variations in shape among their constituent parts, occlusions, and discontinuous surfaces, the issue of missing point cloud data becomes more pronounced compared to regular objects. This is particularly noticeable

✉ Ming-Deng Shi
smdeng@taru.edu.cn

✉ Ya-Jun Xi
ya-junxi@qq.com

[1] College of Information Engineering, Tarim University, Alaer 843300, China

[2] College of Information Science and Engineering, Xinjiang University of Science and Technology, Korla 841000, China

[3] Network Information Center (NIC), Tarim University, Alaer 843300, China

[4] Tarim University Library, Tarim University, Alaer 843300, China

[5] Key Laboratory of Tarim Oasis Agriculture, Ministry of Education, Tarim University, Alaer 843300, China

when dealing with plant point cloud data [9]. Therefore, generating complete point clouds for 3D shapes with morphologically diverse structures is essential for addressing the challenge of missing point cloud data and advancing relevant research in the field [10–13].

Reconstructing an entire object from a partial or incomplete point cloud has become a popular topic of research in recent times. Early approaches in point cloud completion used voxel localization and 3D convolution, adapting established 2D completion techniques to 3D point clouds. In the past few years, researchers have investigated different methods to address the completion of point clouds in deep learning [14–16], such as voxel grids [17], meshes [18, 19], and point clouds [20, 21]. The popularity of 3D analysis based on point clouds has surged following the success of PointNet++ [22], which enables direct processing of 3D coordinates. Encoder–decoder schemes, which have been employed in various pioneering works, have further advanced the field of point cloud completion.

By utilizing an encoder–decoder network, L-GAN [23] was the first to apply a deep learning framework to point cloud completion. Subsequently, PCN (Point Completion Network) [14] integrated the benefits of L-GAN [23] and FoldingNet [15], which focused on incomplete point cloud repair. In addition, PF-Net [24] presented a GAN with a reinforcement learning agent to speed up the point cloud completion prediction time. These methods exhibit remarkable success in restoring the original shapes from incomplete point clouds.

Existing completion datasets are typically generated by sampling the shapes from 3D model datasets [25–27]. However, these datasets often assume that the input point cloud data consist of objects with regular shapes and continuous surfaces, such as cars, tables, and planes. Nonetheless, this paper contend that this assumption may not always hold in real-world scenarios. Consider, for instance, the scanning of a plant in a scene, where the scanning devices inevitably capture a partial 3D shape characterized by diverse structures and discontinuous surfaces. In this context, point cloud completion confronts a practical challenge, as the partial point cloud model to be completed inherently contains dissimilarly shaped structures and discontinuous surfaces, which severely impairs the completion performance. Consequently, the existing methods are generally ill-equipped to handle the completion of partial point clouds exhibiting such morphological diversity and discontinuities.

This paper presents SegCompletion, a novel deep neural network, designed to address the challenge of completing point clouds from a partial 3D shape with diverse structures and discontinuous surfaces encountered in real-world scenarios. First, morphological segmentation is introduced before point cloud completion by deep hierarchical feature learning on point sets, and thus, the complex morphological structure is segmented into regular shapes and continuous surfaces. Second, HDBSCAN [28] is utilized to effectively cluster instances of point clouds belonging to the same feature type. Third, a multiscale generative network is employed to achieve sophisticated patching of missing point clouds based on feature points under the same geometric feature. To account for the variance in mean distances between patch centers and their closest neighbors, a simple yet effective uniform loss is utilized. Results from the experiment proved that SegCompletion was successful in completing point clouds of 3D shapes with different morphologies, as seen in Fig. 1. In summary, the main contributions of this research are outlined as follows.

- We propose SegCompletion, a deep neural network, to tackle point cloud completion from a partial 3D shape with differently shaped structures and discontinuous surfaces in real-world scenarios. The experimental results indicated that SegCompletion could deal with completion in actual circumstances, including diverse and challenging scenarios.
- We suggest the incorporation of a uniform loss to alleviate the discrepancy in mean distances between the patch centers and their respective closest neighbors in the GAN (Generative Adversarial Network). This strategy effectively mitigates the problem of excessive concentration of generated points in the GAN framework by facilitating the separation of the generated points.
- We construct a 3D point cloud dataset of cotton plants, the Cotton3D dataset, which comprises more than 724 high-quality partial and complete point cloud cotton plants. The solution of the missing point cloud data of cotton plants can be beneficial for research related to cotton plants.

This paper is structured as follows: "Related works" gives an introduction to the relevant work on point cloud segmentation and completion, "Method" presents SegCompletion of the network and the loss function in more detail, "Experiments" describes the practical results of the implementation and the experiments on the public datasets, "Discussion" considers further applications of the network in cotton plants, and "Conclusions" concludes the research.

Related works

Utilization of deep learning techniques has been extensive in 3D reconstruction and representation learning, promoting progress in 3D shape completion research, which can be classified into two main approaches [16, 29–32]. (1) Traditional 3D shape-completion methods [33–36] typically rely on hand-crafted features, such as surface flatness or symmetry axes, to estimate the absent parts of incomplete shapes.

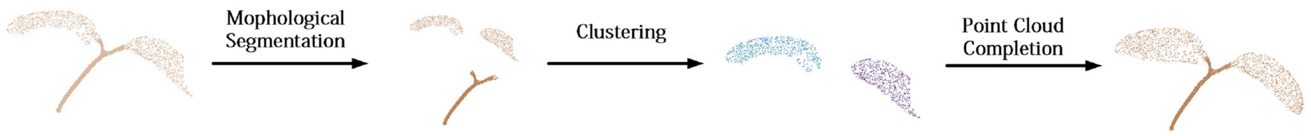


Fig. 1 A partial illustration of SegCompletion results at a variety of different stages

Furthermore, other methods [34, 37–40] utilize large, complete 3D shape datasets to search for similar patches and fill in the incomplete regions. (2) Leveraging the representation learning capacity of deep learning [24, 41–43], incomplete input shapes can be used to extract geometric features, which can then be used to directly infer the complete shape. In contrast to traditional completion methods, these learnable approaches do not require predefined hand-crafted features, allowing them to effectively utilize the abundant shape information present in large-scale completion datasets [32]. Numerous experimental [44] results have shown that these methods perform well with point cloud completion on point cloud datasets with regular shapes and continuous surfaces (e.g., cars, tables, and planes). However, these methods have not yet addressed the challenge of completing point clouds for objects with morphologically diverse structures and discontinuous surfaces. In this section, we primarily survey the research relevant to our work.

3D point cloud segmentation

3D point cloud segmentation is a basic problem in computer vision, and its main task is to output the semantic label value of each point in the 3D scene through a 3D point cloud semantic segmentation algorithm for the given data describing the 3D scene, such as 3D point clouds and color-depth (RGB-D) maps. 3D point cloud semantic segmentation is the basic task of advanced artificial intelligence tasks, such as automatic driving navigation planning and industrial automatic control grasping, and is also the current research hotspot in 3D computer vision and deep learning [45]. Deep learning methods for 3D point cloud semantic segmentation can be classified into Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), Attention Networks, Transformers, and other networks. Attention-based, Transformer-based, and other network-based approaches are all viable options [46].

DGCNN [47] was a Graph Neural Network (GNN)-based algorithm for 3D point cloud segmentation. It dynamically constructs a relationship graph between points to capture local features. GraNet then used GNN to capture local and global features for semantic segmentation, although the computational complexity of GNN could be high. KPConv [48] was an attention-based point cloud convolution method for 3D point cloud segmentation. It used a learnable convolution kernel to dynamically adjust the influence of each point to better capture local and global features, though it might have required more training data for optimal performance. TransPoint [49] was a Transformer-based method for 3D point cloud segmentation. Like Transformers in natural language processing, it used a multi-head self-attention mechanism to capture features in the point cloud. This approach performed well in point cloud segmentation, although it might have also required more computational resources. ShapeNet [50] was a deep learning-based approach for 3D point cloud segmentation. It utilized convolution and aggregation operations to process point cloud data for high-quality segmentation tasks, although it may have required high-quality and integrity of data.

PointNet++ [22], an improved version of PointNet [51], was able to better capture multiscale features, making the segmentation more accurate and comprehensive. Not only that, but it was also applicable to several types of point cloud data, including irregular, sparse, or dense point clouds. In addition, PointNet++ supported end-to-end training, simplifying the training process, and had high memory efficiency to manage large-scale point cloud data. PointNet++ performed well in semantic segmentation tasks, capturing fine-grained semantic information. Although other algorithms also had their merits, PointNet++, as a comprehensive and versatile approach, successfully solved many point cloud segmentation problems and demonstrated impressive performance in practical applications.

In this study, PointNet++ was employed to facilitate the distinction between different morphological features. Consequently, the complex morphological structure was effectively segmented into regular shapes and continuous surfaces.

3D point cloud clustering

Clustering algorithms are a category of unsupervised learning algorithms used to group samples in a dataset based on similarity. In clustering analysis, the measurement of similarity between samples is crucial. The objective of clustering algorithms is to maximize the similarity within the same cluster and minimize the similarity between different clusters. Over the past few decades, researchers have proposed a variety of clustering algorithms, covering diverse methods, and techniques [52].

The K-means clustering algorithm [53] assigned data points to the nearest cluster centroid and updates centroids iteratively to minimize the within-cluster sum of squares.

However, it was sensitive to the initial choice of centroids and required the number of clusters to be predefined. Hierarchical clustering [54] built a hierarchy of clusters by merging or splitting them based on a specified criterion. It offered flexibility, but it could be computationally expensive for large datasets and may produce unstable results. Spectral clustering [55] constructed a similarity graph and performs clustering on the eigenvectors of the graph Laplacian matrix. It captured complex structures but could be computationally demanding, sensitive to parameter selection, and required careful parameter tuning.

Density-based clustering algorithms were commonly used to group data points based on their density, with a minimum number of points within a specified distance considered a cluster. HDBSCAN [28], as a density-based clustering algorithm, offered several notable advantages. It was capable of effectively managing clusters of different shapes and sizes, making it suitable for a wide range of datasets. Additionally, HDBSCAN exhibited robustness in parameter selection, alleviating the need for manual tuning and providing reliable results. These characteristics collectively contributed to HDBSCAN's significance as a valuable clustering algorithm in various domains.

3D shape completion based on GAN

Taking inspiration from the achievements of GANs [56] in 2D tasks such as image repair and processing [57], researchers have made significant advancements in point cloud completion using the traditional GANs, leading to remarkable successes [41].

LGAN [58] introduced a deep generative network for point cloud completion, pioneering its application in this field. However, its architecture was not tailored to shape-completion tasks, resulting in subpar performance. FoldingNet [15] proposed a unique decoding operation called folding, which enables mapping from 2 to 3D. Subsequently, PCN [14] developed a reinforcement learning architecture specifically focused on addressing the challenge of shape completion. PCN was deployed to employ folding to approximate a surface that was smooth and satisfied the completion of the shape. A recent development is a GAN-based network, RL-GANNet [42], which utilized a reinforcement learning agent to enable real-time point cloud completion. An RL agent [59] was incorporated to simplify the optimization process and accelerate prediction, but it did not aim to improve the accuracy of predicting the points. PF-Net [24] was designed to handle partial point cloud input and generate the missing part of the point cloud as output, rather than the whole object. It employed a multiresolution encoder for extracting point cloud features, a PPD (Point Pyramid Decoder) for constructing point clouds, and a GAN discriminator for network refinement. However, the GAN network faced the issue of producing point clouds that exhibited excessive concentration.

In this paper, PF-Net [24] aimed to achieve accurate patching of missing point clouds based on geometric features, demonstrating effective results for various missing rates and multiple missing locations. To ensure surface smoothness and encourage planar representations, a uniform loss [60] was incorporated in the discriminator within the GAN network. Additionally, the average distance between the center of the patch and its nearest neighbor was evaluated to penalize any discrepancies. These modifications contributed to enhancing the quality of the completed point clouds generated by PF-Net.

Method

A graphical representation of the proposed method is presented in Fig. 2. SegCompletion comprises two primary modules: (1) the segmentation and clustering network (Fig. 2a), which is responsible for segmenting and extracting various parts of the morphology, and (2) the point cloud completion network (Fig. 2b), which focuses on filling in missing point clouds and enforcing shape uniformity. A comprehensive explanation of each module is provided in the forthcoming section.

Segmentation and clustering network

The segmentation and clustering network was designed based on PointNet++ [22] in this paper. Given a point cloud set as the input $P = \{x_1, x_2, \ldots, x_n\}$, while $x_i \in \mathbb{R}^d$ and a destination point cloud set $P' = \{x_1', x_2', \ldots, x_n'\}$, with a set function $f : \chi \to \mathbb{R}$. γ is a continuous function, and MAX is a vector max operator that takes n vectors as input and returns a new vector of the element-wise maximum

$$f(x_1, x_2, \ldots x_n) = \gamma\left(\underset{i=1,\ldots,n}{\text{MAX}} \{h(x_i)\}\right). \qquad (1)$$

The paper applied the FPS (farthest point sampling) to select a subset of points $\{x_i^1, x_i^2, \ldots x_i^m\}$, in such a way that x_i^j was the most distant point, which was in the metric distance, from the set $\{x_i^1, x_i^2, \ldots x_i^{j-1}\}$ with respect to the other points. The input was a point set of size $N_l \times (d + C)$ for the number of points N_l with C-dim point features and d-dim coordinates at the set abstraction layer l. This output was a matrix $N_{l'} \times \left(d + C'\right)$ containing subsampled points $N_{l'}$, each with C'-dim feature and d-dim coordinate vectors that summarized the local context, as well as the coordinates of a set of centroids with a size of $N_{l'} \times d$. Each layer l was taken as its input localized groups of points $N_{l'}$ with a data size $N_{l'} \times K \times (d + C)$ that each group related to the local

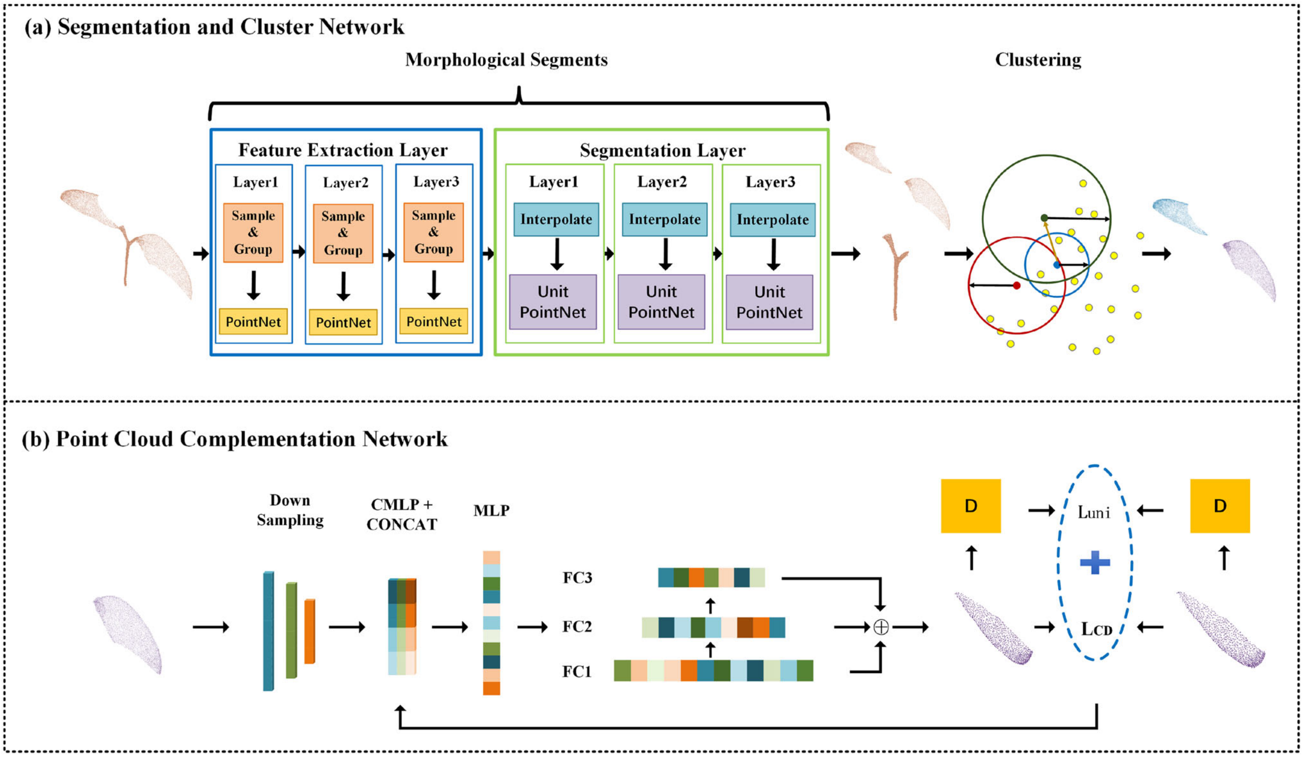


Fig. 2 The detailed structure of SegCompletion. It consists of two parts: **a** the segmentation and clustering network to distinguish between the distinctive features of the morphology by PointNet++. Therefore, each instance of a point cloud belonging to the same category of objects can also be effectively indented using the clustering scheme based on HDBSCAN. **b** A uniform loss is proposed that efficiently penalizes the discrepancy between the average distances of patch centers and their closest neighbors in the PF-Net network

region, and where the number of points was K in the vicinity of the centroid points. Set segmentation was accomplished by the propagation of points from one set to another, known as point feature propagation

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)} \quad \text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p} \quad j = 1, \ldots, C. \tag{2}$$

To avoid the problem of misclassification in morphologically diverse structures, the HDBSCAN method [28] was added to the PointNet++ model to find the full dense region of sample points in this paper. We broke the target point cloud $P' = \{x'_1, x'_2, \ldots x'_n\}$ out into a series of steps.

Transform the space according to the density/sparsity.

As a very inexpensive estimate of density, KNN (Kth Nearest Neighbor) was used and $\text{core}_k(x)$ stood for the core distance between the current point x to its kth closest point

$$\text{core}_k(x') = d\left(x', N^k(x')\right). \tag{3}$$

A new distance metric, the mutual reachability distance, was developed to disperse points that had a low density and a high core distance

$$d_{\text{mreach}-k}(x_i\prime, x_j\prime) = \max\{\text{core}_k(x_i\prime), \text{core}_k(x_j\prime), d(x_i\prime, x_j\prime)\}, \tag{4}$$

where $d(x_i\prime, x_j\prime)$ was the original metric distance between $x_i\prime$ and $x_j\prime$. This metric ensured that dense points, characterized by a low core distance, maintain their relative proximity to each other. In contrast, it forced sparser points to be separated by at least their core distance from any other point, effectively pushing them further apart.

Build the minimum spanning tree of the distance-weighted graph

In this section, the process of constructing the Minimum Spanning Tree (MST) for the distance-weighted graph was described. Initially, the dataset was treated as a weighted graph, where each data point represented a vertex, and the weights of the edges between data points signified their mutual reachability distance. To enhance the theoretical foundation, the section was extended to provide a more comprehensive description of the methodology. To begin, a threshold value was established to govern edge selection. Edges with weights exceeding this threshold were excluded from the graph, ensuring that only the most relevant connections were retained for further analysis

$$E(G) = \left\{\left(x_i, x_j\right) | d\left(x_i, x_j\right) \le \text{threshold}\right\}. \tag{5}$$

Here, $E(G)$ represented the set of edges, x_i and x_j were vertices in graph G, $d\left(x_i, x_j\right)$ represented the distance between vertices x_i and x_j, and threshold was the predefined threshold value.

Subsequently, the Prim algorithm, a classical method for finding the MST of a weighted graph, was applied. This algorithm efficiently and systematically identified the edges that constituted the Minimum Spanning Tree while avoiding cycles and redundancies.

Build the cluster hierarchy

To convert the minimal spanning tree into a hierarchy of connected components, the edges of the tree needed to be sorted in ascending order based on their distances. As the edges were traversed, it was important to identify the two clusters that would be joined together by each edge. This was achieved by employing a union-find data structure, which allowed for the management of cluster connectivity. By utilizing this approach, the merging of clusters was accurately determined, while the edges of the tree are being traversed.

Let G be the weighted graph representing the data points, where V represents the set of vertices (data points) and E represents the set of edges with their corresponding weights. The union-find data structure UF is utilized to maintain the disjoint sets representing clusters.

For each edge (u, v) in ascending order of edge weights $w(u, v)$:

1. If UF.Find(u) $\neq$ UF. Find(v), merge the clusters UF.Find(u) and UF.Find(v).
2. Update the cluster hierarchy to reflect the merging of clusters.

Condense the cluster hierarchy based on the minimum cluster size

To determine which clusters should be split and which should remain intact, a minimum cluster size setting was introduced in this study. This method involves traversing the cluster hierarchy while providing precise control over cluster behavior. Specifically, a threshold for the minimum cluster size, denoted as minSize, was defined. During the traversal of the cluster hierarchy, the following process was implemented:

(1) If a split resulted in the creation of new clusters with a size smaller than minSize, the larger cluster retained its integrity, and the data points separated from it were marked, with their corresponding distance values recorded

$$\text{if}(|C_i| < \text{minSize}),\ C_j = C_j \cup C_i,\ C_i = \emptyset, \tag{6}$$

where C_i represents a cluster with a size less than minSize, and C_j is a larger cluster.

(2) Conversely, if a split led to the formation of two new clusters, both of which were equal to and larger than minSize, the split was allowed to proceed.

Through this process, a more streamlined hierarchy was obtained, providing insights into how cluster sizes decrease with varying distances. This enhancement enriches the content and theoretical depth of the method section and can be succinctly expressed using mathematical formulas.

Extract the clusters

A different measure than distance $\lambda = \frac{1}{\text{distance}}$ was to consider the persistence of clusters. For each cluster, the stability was computed as

$$S_{\text{cluster}} = \sum_{p \in \text{cluster}} \left(\lambda_p - \lambda_{\text{birth}}\right). \tag{7}$$

The lambda values λ_{birth} and λ_{death} denoted the instances when a cluster had split off and formed its own separate cluster and when a cluster had divided into smaller clusters, respectively. Each point p within a cluster had an associated lambda value λ_p, representing the moment when the point “exited the cluster”. This transition occurred between λ_{birth} and λ_{death}, as the point either left the cluster during its existence or departed when the cluster underwent further subdivision into smaller clusters.

In the reverse topological order traversal of the tree, all leaf nodes were considered as individual clusters. The stability of each cluster was determined based on the sum of the stabilities of its child clusters. If the sum of the child stabilities surpassed the stability of the cluster, the cluster’s stability was updated to the sum of the child stabilities. Conversely, if the cluster’s stability was higher than the sum of its children, the cluster was marked as selected, and all its descendant clusters were unselected. This process continued until the root node was reached. The resulting set of selected clusters at this stage represented the flat clustering, which was then returned as the final outcome.

Point cloud completion network

This research utilized PF-Net [24] for point cloud completion, which was made up of two deep networks, namely a discriminator D and a generator G. The generator G produced artificial examples, while the discriminator D attempted to differentiate between real and fake samples from the entire dataset. The first-order approximation of multilayer graph convolution with Chebyshev expansions was introduced as follows:

$$p_i^{l+1} = \sigma\left(Y_i^{l+1} + \sum_{q_j \in A\left(p_i^l\right)} U^l q_j^l + b^l\right), \tag{8}$$

where p_i^l was the ith node at the lth layer of the complementary network, q_j^l was the jth neighbor of p_i^l, and the set $A\left(p_i^l\right)$ consisted of all ancestors of p_i^l. In the training process, GCNs determined the most suitable weights U^l and bias b^l at each layer and used these parameters to generate 3D coordinates for point clouds, thus ensuring their resemblance to real point clouds. K backing from F_K^l was responsible for the production of Y_i^{l+1}, where F_K^l was a fully connected layer containing K nodes. $N\left(p_i^l\right)$ was the set of all neighbors of p_i^l. $\sigma(\cdot)$ was the activation unit. The following point was proposed by the new loop term based on K support as follows:

$$Y_i^{l+1} = F_K^l\left(p_i^l\right). \tag{9}$$

Training loss

Regularization of the completion shape was achieved by the complete ground-truth point cloud, with the help of uniform loss and CD (Chamfer Distance) [61]. The CD, being the most widely implemented structural loss for shape completion, was comparatively insensitive to details and density distribution [17, 62]. A uniform loss could be employed to rectify the problem of point cloud generation being unevenly distributed. Equation (10) defined the loss function of a discriminator $\mathcal{L}$

$$\mathcal{L} = \mathcal{L}_{\text{com}} + \mathcal{L}_{\text{uniform}}. \tag{10}$$

Uniform loss: To resolve the problem of irregularly distributed point cloud creation, a uniform loss [63] that should improve the generator's generative ability was used. The uniform loss $\mathcal{L}_{\text{uniform}}$ was defined as

$$\mathcal{L}_{\text{uniform}} = \text{Var}\left(\left\{\rho_j\right\}_{j=1}^n\right), \quad \rho_j = \frac{1}{k}\sum_{i=1}^k \text{dist}^2\left(y_i, y_j\right). \tag{11}$$

Specifically, the paper randomly selected n seed positions on the object surface using the FPS method, and then, small patches were formed by incorporating the k-nearest neighbors of every seed. These small patches exhibited similar scattering, regardless of whether the structure was fine or coarse. Thus, the average distance from each seed to its k-nearest neighbors was calculated, and the variance of the average distances of all patches was penalized as expressed in Eq. (9).

Chamfer distance: The CD does not qualify as a distance function due to its deviation from the triangle inequality [61]. In spite of this, the phrase "distance" was utilized to signify any nonnegative function that is established on pairs of point sets. The CD looked for the nearest point y in the other set $\hat{Y}$ for each point x in point set Y, computed the squared distances, and repeated the process in the opposite direction. The CD was seen as a continuous and piecewise smooth function when viewed in terms of the point locations in Y and $\hat{Y}$. Each point's range search was executed in parallel, as they were independent of each other. The $\mathcal{L}_{CD}$ between Y and $\hat{Y}$ was a metric used to calculate the separation between them

$$\mathcal{L}_{\text{CD}} = \text{CD}\left(Y, \hat{Y}\right) = \sum_{x \in Y} \min_{y \in \hat{Y}} \|x - y\|_2^2 + \sum_{y \in \hat{Y}} \min_{x \in Y} \|x - y\|_2^2. \tag{12}$$

The multistage completion loss, which followed the point pyramid decoder's prediction of three layers at different resolutions, was stated in Eq. (9) through three terms, d_{CD1}, d_{CD2}, and d_{CD3}, weighted by hyperparameter α. The initial formula d_{CD1} computed the squared distance between the points of the first layer Y_1 and the actual values of the missing region $\hat{Y}_1$. The distance in the square between the points of the second layer Y_2 and the ground-truth point $\hat{Y}_2$ at the missing region was denoted by d_{CD2}. The third expression d_{CD3} computed the squared distance between the points of the third layer Y_3 and the points of the ground truth $\hat{Y}_3$ at the missing region. We obtained $\hat{Y}_2$ and $\hat{Y}_3$ from $\hat{Y}_1$ by applying FPS. The multistage completion loss designed amplifies the number of feature points, resulting in a more precise focus on them

$$\mathcal{L}_{\text{com}} = d_{\text{CD1}}\left(Y_1, \hat{Y}_1\right) + \alpha d_{\text{CD2}}\left(Y_2, \hat{Y}_2\right) + 2\alpha d_{\text{CD3}}\left(Y_3, \hat{Y}_3\right). \tag{13}$$

Cross-entropy loss: As part of the training process, the point cloud segmentation and clustering network works to reduce its cross-entropy loss, the training loss function used

$$H(g, p) = -\sum_{i=1}^n g(x_i)\log(p(x_i)), \tag{14}$$

where $H(g, p)$ was used to measure the discrepancy between the true probability distribution $g(x_i)$ of point clouds and the predicted probability distribution $p(x_i)$ by the model.

A lower value of $H(g, p)$ indicated better performance in model predictions. Here, n represented the number of point clouds.

Experiments

To facilitate a comprehensive evaluation, this paper used the benchmark datasets ShapeNet and Pheno4D. (1) ShapeNet [26]: the CAD (Computer Aided Design) dataset obtained from PCN consists of 30,974 models of three dimensions divided across 8 categories. The point clouds of the ground truth were composed of 16,384 points that were evenly spread over the surfaces. (2) Pheno4D [64]: the dataset comprised 7 tomato plants that were measured over a span of 20 days, resulting in a total of approximately 350 million points. This dataset consisted of 140 points, out of which 77 points (with 200 million points) were annotated with labels. It is important to note that temporally consistent labels were provided for each point within the point clouds.

This paper compared SegCompletion with several representative methods that directly operated on 3D point clouds, namely FoldingNet [15], GRNet [65], PF-Net [24], PMP-Net [20], PMP-Net++ [32], DeCo [66], and SnowflakeNet [67]. All these existing methods were evaluated on different datasets, while we conducted training and testing on the same dataset for quantitative evaluation. To align with previous studies, we utilized the per-point L1 Chamfer Distance [14] and the per-point L2 Chamfer Distance [67] on the testing set for evaluation purposes.

Detailed settings

PointNet++ and HDBSCAN were employed as the foundational framework for the segmentation and clustering network. The segmentation results were notably influenced by the number of output channels in both the encoder and decoder of MLPs (Multi-Layer Perceptron). A detailed description of the architecture for each component is provided in Table 1.

HDBSCAN was employed to effectively segment each point cloud instance that was associated with the same object category. According to the shape of the object, we selected the same parameter value for min_cluster_size = 400 on the ShapeNet, Pheno4D, and Cotton3D datasets (Online Appendix Tables 1–4).

The point cloud completion network was implemented on PyTorch. The ADAM optimizer was employed to alternately train all the building blocks with a batch size of 16, a learning rate of 0.001, and an epoch size of 100. The training process was accomplished using 4 NVIDIA GTX 2080TI GPUs, CUDA 11.6, Ubuntu 22.04.1 LTS, and Python 3.7.

Point cloud completion on the Pheno4D dataset

Point cloud preprocessing

Down-sampling and up-sampling: The experiments demonstrated the effective training of SegCompletion on sparse point clouds. As part of the preprocessing, the training of SegCompletion on sparse shapes with 1024 points was achieved by down-sampling and up-sampling techniques. To account for varying numbers of points, shapes with more than 1024 points were reduced to 1024 using the FPS method. Conversely, shapes with fewer than 1024 points were augmented by randomly replicating the points at different scales to reach the same number.

Point cloud labels: Supervised deep learning techniques necessitated labeled data for training the networks. In this study, the segmentation of tomato plant point clouds for annotation was conducted using CloudCompare software, a point cloud visualization tool. A binary classification approach was adopted, where one color was assigned to represent the tomato leaves, while another color represented the other organs of the tomato plant. A total of 220 plants at the tomato seedling stage were labeled, with 200 of them allocated for training, 12 for validation, and 8 for testing. The ratio between these three subsets was 50:3:2.

Incomplete point cloud: Since SegCompletion focused on point movement rather than generation, the incomplete and complete point clouds needed to have an equal number of points. In the process, the point cloud data were centered at the origin and normalized to the range of [− 1, 1] coordinates. The ground-truth point cloud data were created by uniformly sampling 1024 points from each shape. To generate incomplete point cloud data, a central point was randomly chosen from multiple viewpoints, and points within a certain distance from the complete data were eliminated.

Quantitative comparison

SegCompletion showed a performance that was comparable to the existing state-of-the-art techniques [15, 20, 24, 32, 65–67] and was the highest ranked on the Pheno4D dataset when evaluated using the metrics L1 and L2. Table 2 reveals that the latest methods for point cloud completion, such as FoldingNet, GRNet, PF-Net, PMP-Net, PMP-Net++, DeCo, and SnowflakeNet, have been developed in recent years. Notably, SegCompletion achieved an average L1 of 0.397, which was significantly lower than FoldingNet's average L1 of 12,391.682. Moreover, SegCompletion demonstrated its strong generalization performance by obtaining the lowest L1 results in all categories, indicating its capability in completing shapes of multiple categories. The L2 metric can also be used to support this conclusion.

Table 1 Detailed structure of the encoder and decoder

Level	Down-sampled points	Radius of ball query	Neighbors' point	MLPs (encoder)	MLPs (decoder)
1	512	0.1	32	[32, 64]	[256, 256]
		0.2	64	[64, 64, 128]	
		0.4	128	[64, 96, 128]	
2	128	0.4	64	[128, 128, 256]	[256, 128]
		0.8	128	[128, 196, 256]	
3	–	–	–	[256, 512, 1024]	[128, 128]

Table 2 Point cloud completion on the Pheno4D dataset in terms of per-point L1 and L2 Chamfer distance × 10^{-3} (lower is better)

Methods	L1	L2
PMP-Net [20]	15.564	2.301
PMP-Net++ [32]	14.881	1.945
GRNet [65]	46.911	38.431
SnowflakeNet [67]	16.890	1.633
FoldingNet [15]	12,391.682	3491.486
PF-Net [24]	14.810	2.898
DeCo [66]	15.965	3.512
Our method	**0.397**	**0.014**

The bold values indicate instances where our method shows superior performance

The FoldingNet method excels in handling global features of point clouds, but it exhibits relatively weaker capabilities in capturing local geometric details. While GRNet improves the fusion of global and local features to some extent, it still faces challenges related to point cloud sampling and sparsity. PF-Net focuses on reconstructing partial point clouds but may be sensitive to complex structures and noise. PMP-Net and PMP-Net++ introduce graph context information but encounter challenges of excessive smoothing, leading to the loss of details in reconstructing point clouds. DeCo introduces a decoupling mechanism in the encoder–decoder structure but may have limitations in handling point clouds with multimodal features. SnowflakeNet proposes a multi-level point cloud pyramid but may face difficulties in adapting to diverse shapes. Comparatively, SegCompletion also applied a GAN network for multiscale feature extraction but yielded significantly better results when applied to the Pheno4D dataset. This improvement was attributed to the segmentation and clustering network adopted in SegCompletion, which was capable of distinguishing between the different features of the morphology.

Qualitative comparison

To further highlight the superiority of SegCompletion over other methods, Fig. 3 presents a visual comparison of the overall performance of the Pheno4D dataset. This study aimed to evaluate SegCompletion in comparison to other techniques in terms of their ability to accurately predict the complete shape of a partial point cloud. The results showed that the PMP-Net, PMP-Net++, SnowflakeNet, and GRNet methods generated insufficient complementary points, and the FoldingNet method produced many noise points, resulting in a deformed output. GRNet primarily focused on local features and often performed poorly when dealing with global features and structures. Additionally, its generalization performance on different datasets or point clouds with varying characteristics was limited (Fig. 3). In contrast, SegCompletion demonstrated superior visualization of point cloud completions across various object categories.

SegCompletion stands out from the generative methods, showing its potential to generate a high-quality leaf. By incorporating a segmentation and cluster network into the PF-Net network, the paper observed a marked improvement in the completeness of the Pheno4D dataset. Compared to PF-Net and DeCo, Fig. 4 reveals that SegCompletion was more successful in completing the shape of all leaves. PF-Net focuses on the reconstruction of partial point clouds, but it may be sensitive to complex structures and noise, leading to some defects in the process of shape completion and visualization. DeCo introduces a decoupling mechanism but may face limitations in handling object shape completion, affecting the visualization results. These issues may include unnatural deformations or missing parts when completing object shapes and the presence of artifacts or detail loss in the visualization process.

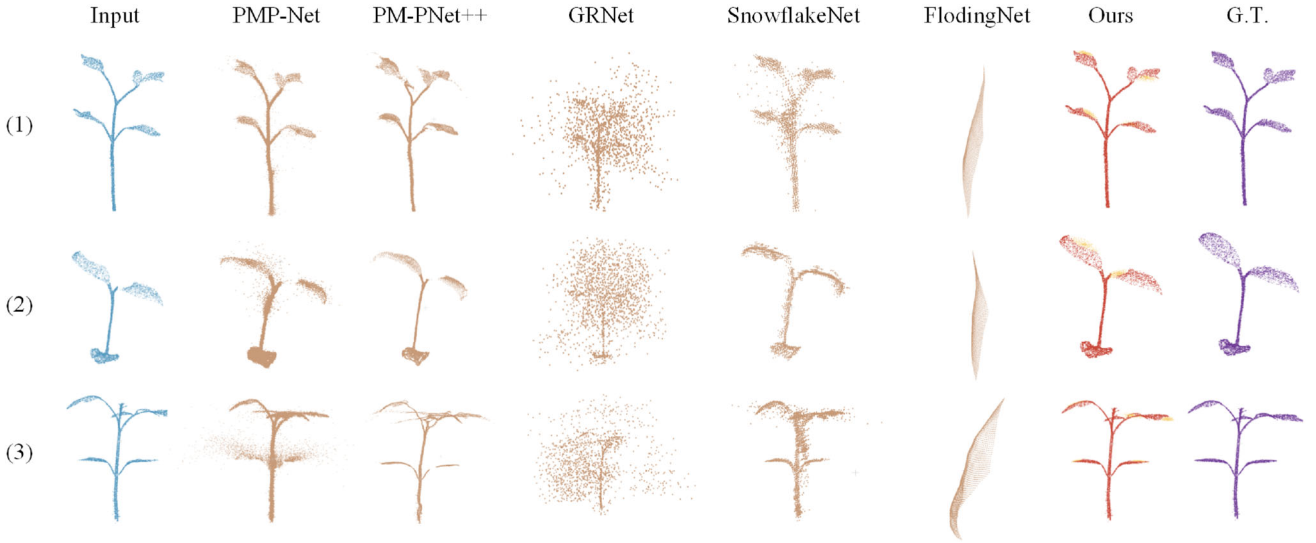


Fig. 3 Visualization of the Pheno4D dataset showing how point cloud completion compares to previous methods. The PMP-Net, PMP-Net++, SnowflakeNet, GRNet, and FoldingNet methods output the whole object, but our method only displays the part of the point cloud that is missing (yellow) rather than the whole object

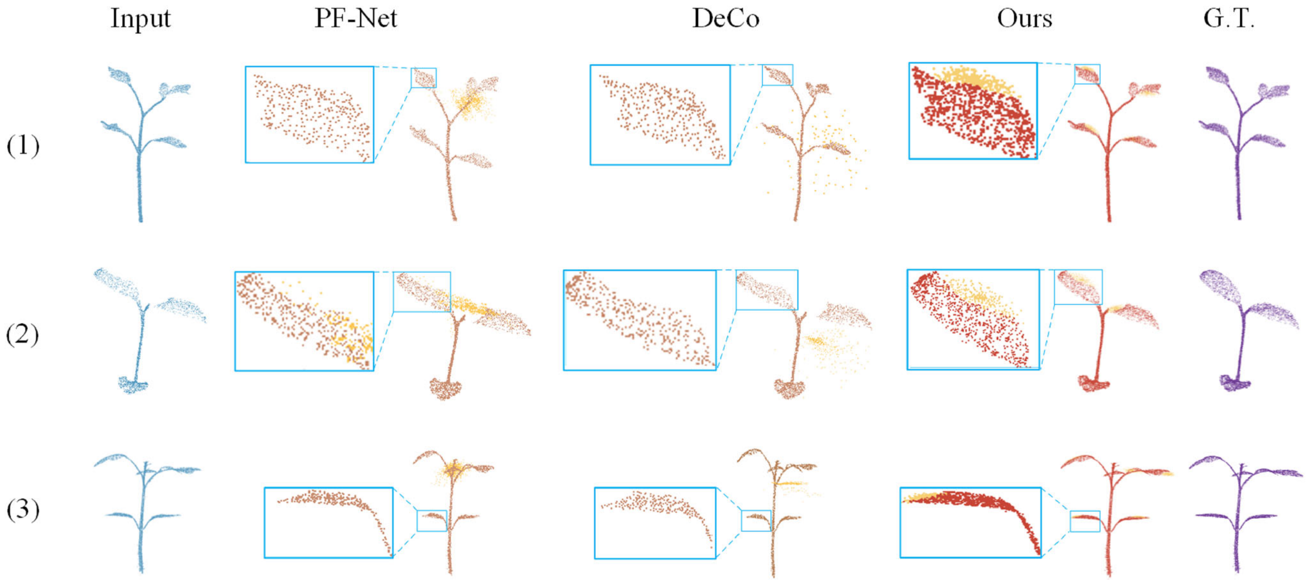


Fig. 4 A visualization showcases how PF-Net, DeCo, and SegCompletion affect the incomplete section of the complement's point cloud

Point cloud completion on the ShapeNet dataset

Point cloud preprocessing

In the ShapeNet dataset, the additional 2048 points were down-sampled to 2048 using the FPS method, while the point clouds with fewer than 2048 points were up-sampled to 2048 by replicating their neighboring points. For each object, the parts with missing data were manually labeled in one color, while the complete parts were labeled in another color. The annotation process was conducted using CloudCompare software, where experts carefully marked the missing parts and labeled the complete parts. The same approach as in the Pheno4D dataset was employed for generating the missing point clouds.

Quantitative comparison

The ShapeNet dataset reveals that the guitar exhibits the most intricate and diverse geometry, with 254 morphologies, while

Table 3 Point cloud completion on the ShapeNet dataset in terms of the per-point L1 Chamfer distance × 10^{-3} (lower is better)

Methods	Average	Plane	Chair	Guitar	Laptop	Mug
PMP-Net [20]	5.502	3.062	7.859	3.956	6.697	5.936
PMP-Net++ [32]	6.017	4.014	5.657	3.634	5.049	11.731
GRNet [65]	16.762	13.021	17.221	10.051	19.903	23.614
SnowflakeNet [67]	14.9774	10.631	13.93	8.219	15.388	26.719
FoldingNet [15]	61.1642	45.391	59.51	38.556	46.514	115.851
PF-Net [24]	5.155	4.331	5.627	2.216	5.652	7.949
DeCo [66]	7.859	10.075	7.932	4.409	7.151	9.729
Our method	**2.96**	**2.696**	**3.336**	**1.329**	**2.813**	**4.626**

The bold values indicate instances where our method shows superior performance

Table 4 The per-point L2 Chamfer distance ×10^{-3} (lower is better) is used to evaluate point cloud completion from the ShapeNet dataset

Methods	Average	Plane	Chair	Guitar	Laptop	Mug
PMP-Net [20]	0.951	0.166	0.464	0.236	0.260	3.629
PMP-Net++ [32]	0.268	0.161	0.400	0.068	0.147	0.562
GRNet [65]	0.546	0.316	0.597	0.198	0.653	0.967
SnowflakeNet [67]	0.698	0.363	0.578	0.315	0.596	1.637
FoldingNet [15]	10.566	5.796	11.559	3.151	5.219	27.107
PF-Net [24]	0.839	0.605	0.896	0.103	0.859	1.734
DeCo [66]	3.039	5.222	3.975	1.730	1.901	2.370
Our method	**0.256**	**0.125**	**0.401**	**0.052**	**0.207**	**0.497**

The most satisfactory outcomes are emphasized with bold font

the notebook has the simplest morphology, with only one form. Table 3 shows that SegCompletion achieves a lower L1 value of 1.329 for guitar geometry completeness compared to state-of-the-art techniques [15, 20, 24, 65, 67]. Additionally, SegCompletion obtained an L1 value of 2.813 for the notebook, which was lower than the PF-Net calculation of 5.652. These results demonstrated that SegCompletion was more effective in achieving completeness for both complex and simple objects in the public dataset. SegCompletion demonstrated its robustness in completing shapes across multiple categories by achieving the lowest L1 values in all categories. Moreover, PF-Net employed a feature-point-based multiscale generating network, enabling hierarchical estimation of the missing point cloud and showcasing its efficacy in various challenging point cloud completion tasks. By utilizing a GAN network for multiscale feature extraction, SegCompletion significantly outperformed other methods when applied to the ShapeNet dataset. This improvement could be attributed to the segmentation and clustering network used in SegCompletion, which effectively recognizes distinct features of different morphologies.

SegCompletion, based on the per-point L2 Chamfer distance, yielded results that were consistent with the per-point L1 Chamfer distance on the ShapeNet dataset. The mean L2 distance was lower than that of the other methods, and it was lower for all five classes of objects. PMP-Net, PMP-Net++, GRNet, SnowflakeNet, and FoldingNet completed the whole object but had varying degrees of error in the reconstruction procedure, except for the missing part. However, PF-Net and DeCo generated too many concentrated points in the absent portion. Table 4 presents that SegCompletion is successful in producing point clouds with increased accuracy and less distortion, both in the complete point cloud and in the point cloud of the missing region.

Qualitative comparison

To further highlight the advantage of SegCompletion over other approaches, Fig. 5 presents a visual comparison of its performance on the ShapeNet dataset. The objective of this study was to compare SegCompletion with alternative methods across various object categories. For instance, in Fig. 5, the second row illustrates the prediction of the complete shape of a partially visible laptop. Many of the evaluated methods encountered difficulties in accurately preserving the precise geometries of the laptop screen. The PMP-Net and SnowflakeNet methods generated an insufficient number of complementary points, while the GRNet method introduced excessive noise points. Additionally, the results obtained from the FoldingNet method exhibited significant distortions.

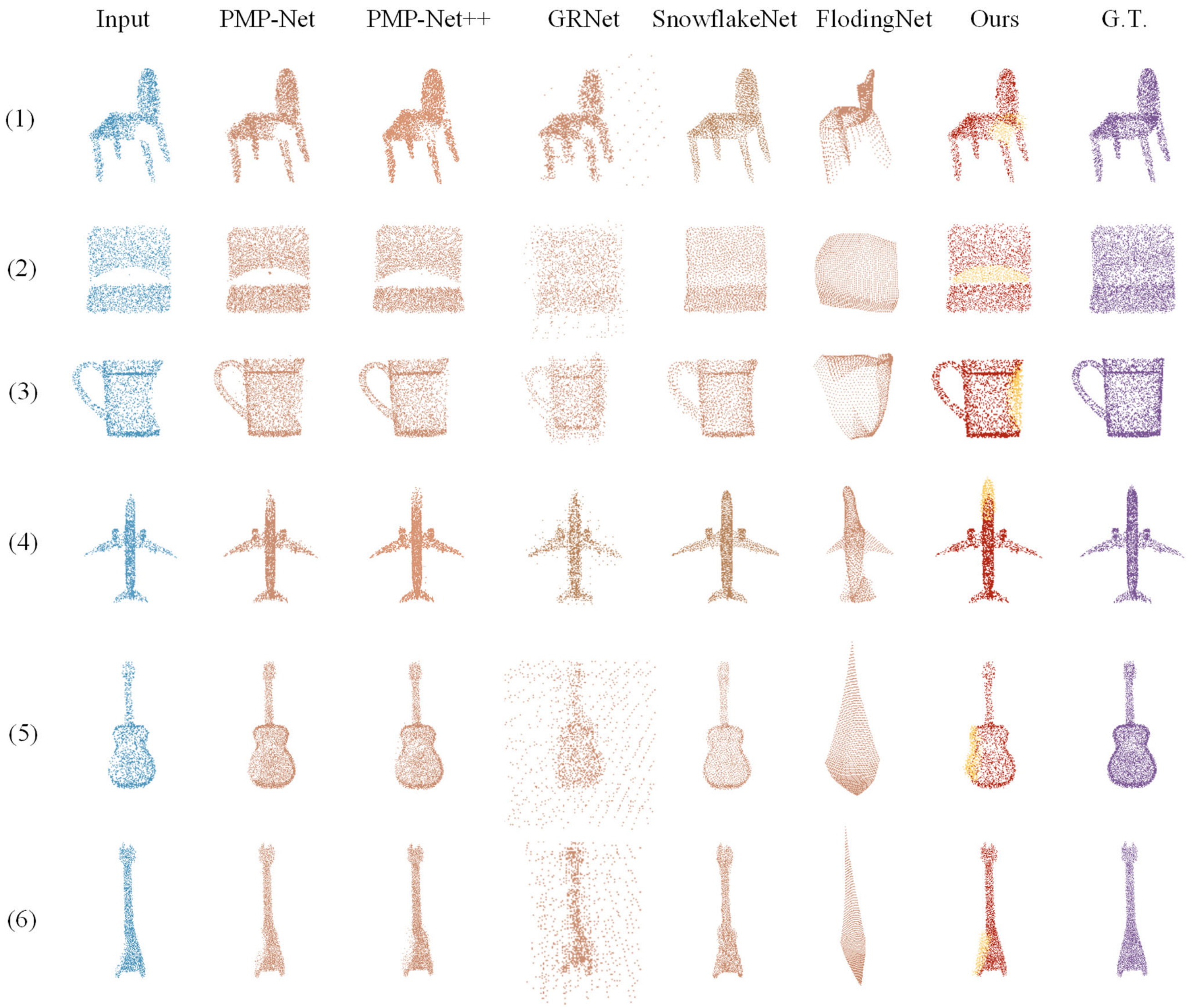


Fig. 5 Visualization of the ShapeNet dataset showing how point cloud completion compares to previous methods. The PMP-Net, PMP-Net++, SnowflakeNet, GRNet, and FoldingNet methods output the whole object, but our method only displays the part of the point cloud that is missing (yellow) rather than the whole object

In contrast, SegCompletion outperformed these methods by producing more accurate and visually appealing point cloud completions. Its ability to preserve precise geometries and avoid excessive noise points distinguished it from PMP-Net, SnowflakeNet, GRNet, and FoldingNet. Thus, SegCompletion demonstrated its superiority in visualizing point cloud completions across diverse object categories.

PF-Net and DeCo excelled in preserving the spatial configuration of the partial point cloud and accurately inferring the intricate geometric structure of the missing areas during the prediction process. However, they tended to prioritize shape completion over other aspects. In contrast, SegCompletion stands out among the generative methods presented in Fig. 6, demonstrating its ability to deliver high-quality reconstructions of laptops. By incorporating a segmentation and clustering network into the PF-Net architecture, SegCompletion significantly improved the complementary effect on the ShapeNet dataset. Moreover, when compared to PF-Net, the results suggested that SegCompletion was more effective in accurately predicting the shapes of objects across all categories than other methods.

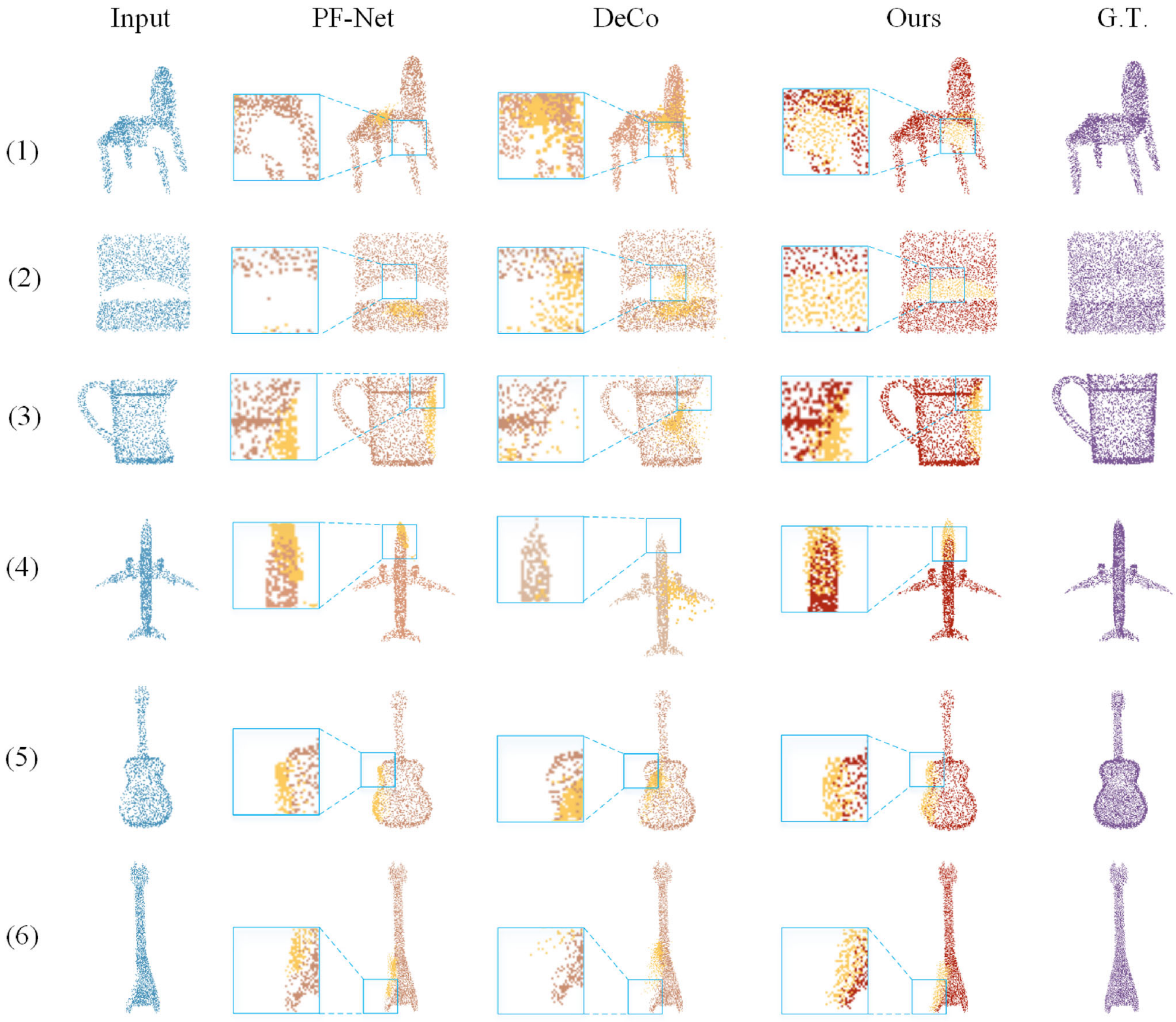


Fig. 6 A visualization displays how PF-Net, DeCo, and SegCompletion affect the incomplete section of the complement's point cloud

Ablation studies

Effectiveness of each component

The impact of the segmentation and clustering module and the uniform loss in the SegCompletion method was evaluated by investigating their efficacy when they were removed. Four different variants were designed for comparison: (1) no-segmentation, where the segmentation and clustering unit was omitted from the network; (2) no-uniform loss, which excluded the uniform loss component from the network; (3) PF-Net, where both the morphological segments and the uniform loss were removed; and (4) the full model, representing the complete SegCompletion method.

Table 5 presents compelling evidence that the full SegCompletion model achieves superior results compared to other network variants. The comparison between the No-Segmentation model and the full model highlights the value of incorporating the segmentation and clustering module before network completion. Similarly, the comparison between the No-Uniform Loss model and the full model demonstrates the effectiveness of addressing the issue of nonuniformity in shape point clouds. Moreover, the comparison between PF-Net and the full model emphasizes the significant contribution of the combined modules to the overall performance.

Table 5 Point cloud completion performances of no-segment, no-uniform, PF-Net, and the full model on the ShapeNet dataset in relation to the per-point L2 Chamfer distance (lower is better, × 10^{-3})

	Average	Guitar	Mug	Chair
No-segmentation	0.379	0.079	0.601	0.457
No-uniform loss	0.667	0.091	1.142	0.769
PF-Net	0.911	0.103	1.734	0.896
Full model	**0.317**	**0.052**	**0.497**	**0.401**

The bold values indicate instances where our method shows superior performance

Table 6 The partial point cloud is reduced by 25%, 37.5%, 50%, and 75% of the initial point cloud

Missing rate	25%	37.5%	50%	75%
Pred → GT error	0.014	0.025	0.036	0.054
GT → Pred error	0.026	0.038	0.084	0.094
L2	0.052	0.064	0.116	0.128

The metric is the error between Pred and GT with respect to L2 (lower is better, × 10^{-3})

Robustness to the model

The robustness tests conducted in this study were focused on the "Guitar" class. To assess the reliability of SegCompletion, a robustness test was performed by controlling the number of output points and training the model to repair shapes with varying degrees of incompleteness. The experimental results are summarized in Table 6. In comparison to the ground truth, percentages of 25, 37.5, 50, and 75 indicated that four partial inputs were missing 512 points, 768 points, 1024 points, and 1536 points, respectively. The robustness of SegCompletion was evident in the similarity between the errors of the predicted (Pred) and ground truth (GT) for the four partial inputs. This indicated that SegCompletion was able to effectively manage inputs with different levels of incompleteness. The findings demonstrated that SegCompletion exhibited robustness in accurately completing point cloud shapes, even in the presence of significant missing data. This highlights its potential applicability in scenarios where incomplete or partially scanned point clouds are encountered.

The effectiveness of SegCompletion on the test set is illustrated in Fig. 7, highlighting its ability to accurately distinguish between distinct types of guitars while preserving the intricate geometric details of the original point cloud. This holds even when dealing with substantial levels of incompleteness. To further evaluate the robustness of SegCompletion, a second test was conducted. In this test, the model was trained to complete partial shapes by filling in multiple missing points located at unusual positions. The purpose of this test was to assess SegCompletion's performance in handling incomplete inputs with varying degrees of incompleteness.

Table 7 Point cloud completion on the Cotton3D dataset in terms of the per-point L1 and L2 Chamfer distance ×10^{-3} (lower is better)

Methods	L1	L2
PMP-Net [20]	22.605	0.485
PMP-Net++ [32]	18.681	1.633
GRNet [65]	47.689	1.421
SnowflakeNet [67]	15.219	0.866
FoldingNet [15]	112,958.212	14,881.678
PF-Net [24]	19.384	0.146
DeCo [66]	19.515	4.439
Our method	**6.071**	**0.022**

The bold values indicate instances where our method shows superior performance

Discussion

Completion in cotton plant leaves

The Cotton3D dataset is distinct from the existing datasets due to its various data features and complexities. LiDAR, laser scanners, stereo cameras, RGB-D scanners, and other devices can be used to capture point cloud data of cotton plants, preserving the initial geometric information in three-dimensional space. However, due to the variations in shape, mutual occlusion, and discontinuous surfaces, there is a more severe issue of missing point cloud data when compared to the conventional objects. This study aims to ensure that the SegCompletion method can cover various data scenarios, allowing for a deeper exploration of its generalization capabilities.

The analysis of the data presented in Table 7 reveals significant findings. PF-Net [24] secures the second position with

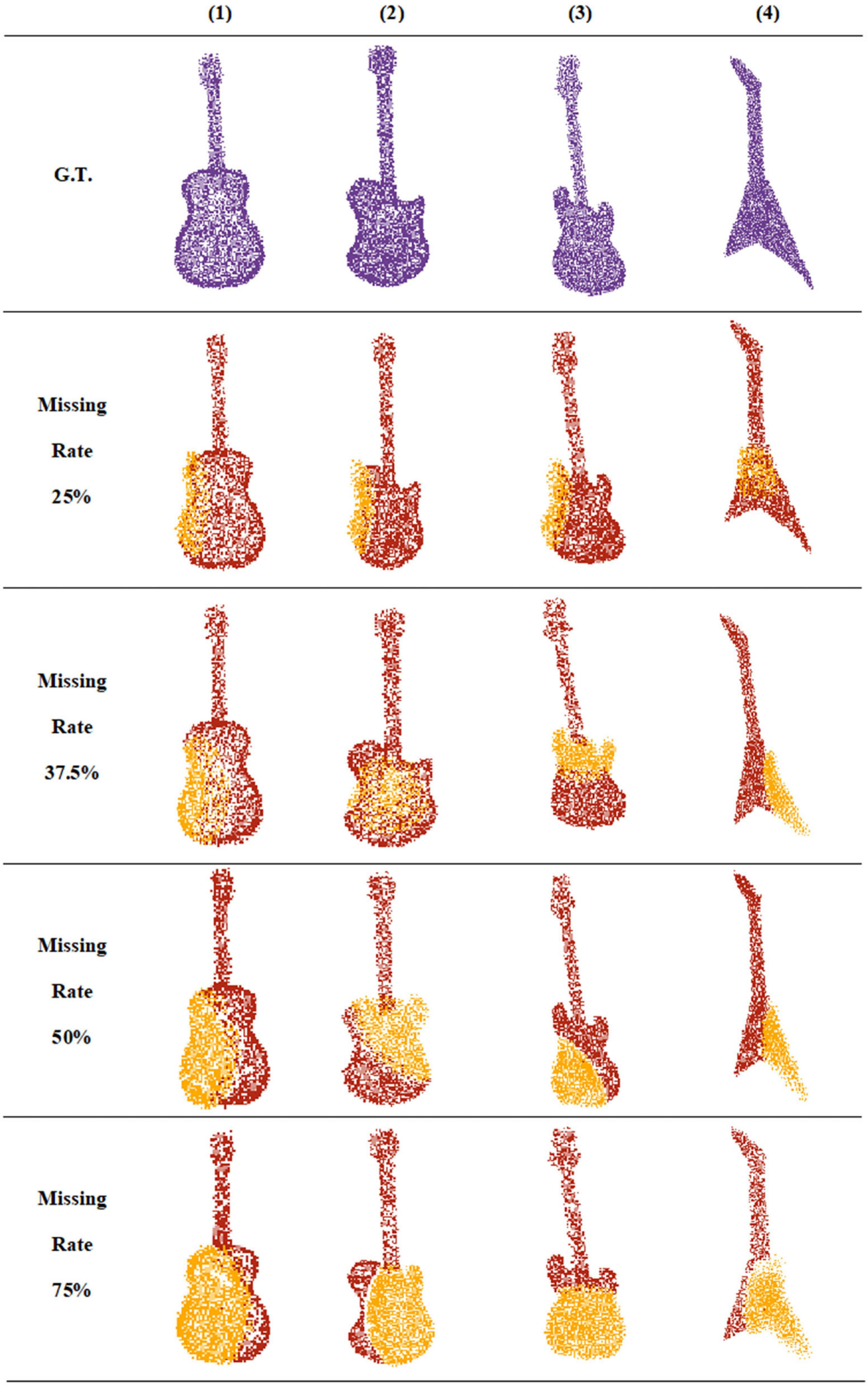


Fig. 7 Examples of repair results when the original input has various degrees of incompleteness. The original point cloud is reduced by 25%, 37.5%, 50%, and 75% for (1), (2), (3), and (4), respectively. Yellow indicates a prediction. Red indicates that the point cloud is not damaged . Purple represents the target ground truth

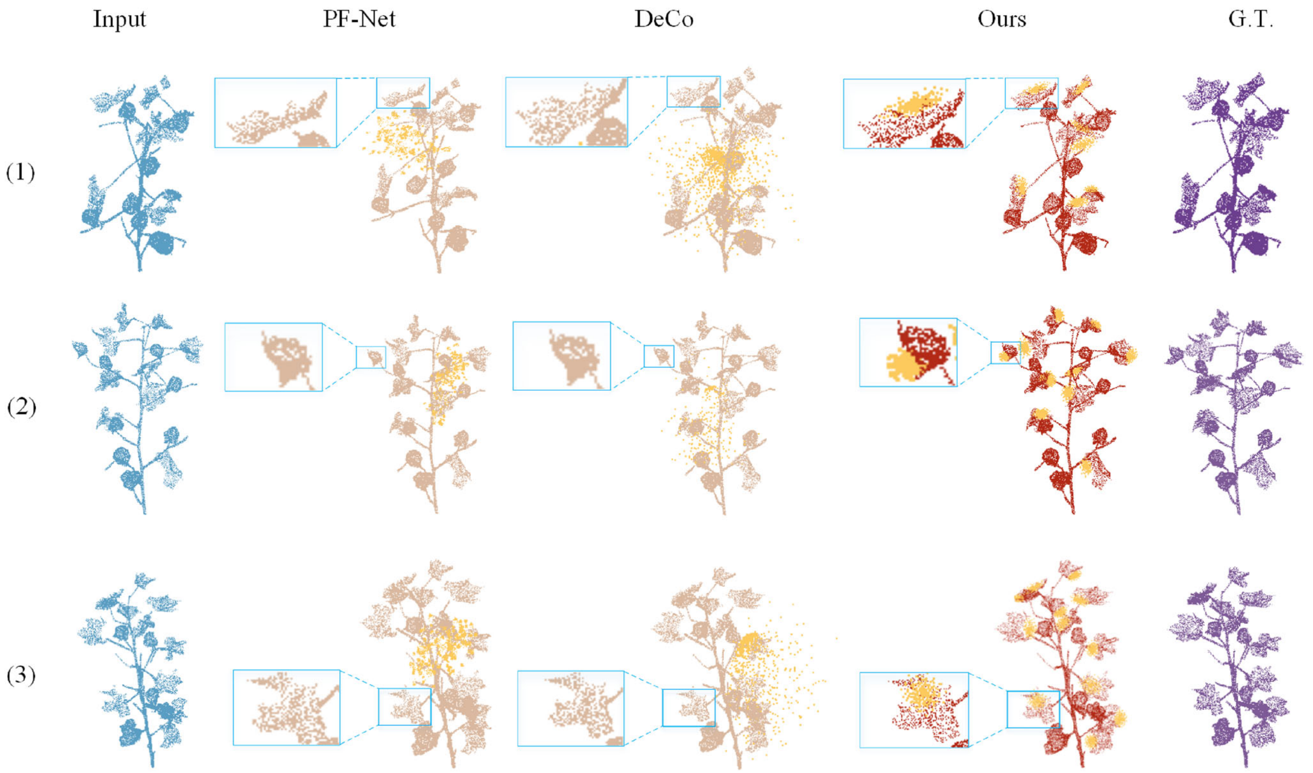


Fig. 8 Visualization of the Cotton3D dataset showing how point cloud completion compares to the PF-Net and DeCo methods

an L2 value of 0.146. In contrast, SegCompletion outperforms PF-Net with a significant improvement of 0.124 in the L2 metric. Notably, on the Cotton3D dataset, SegCompletion demonstrates a remarkable 84.93% reduction in the L2 metric compared to PF-Net. This noteworthy decrease highlights the effective utilization of the abundant geometric information embedded in point clouds by SegCompletion. Furthermore, the superior performance of SegCompletion over SnowflakeNet [67] provides compelling evidence of its effectiveness. It highlights exceptional generalization capabilities when dealing with morphologically diverse structures, surpassing other methods in performance across all cotton plants.

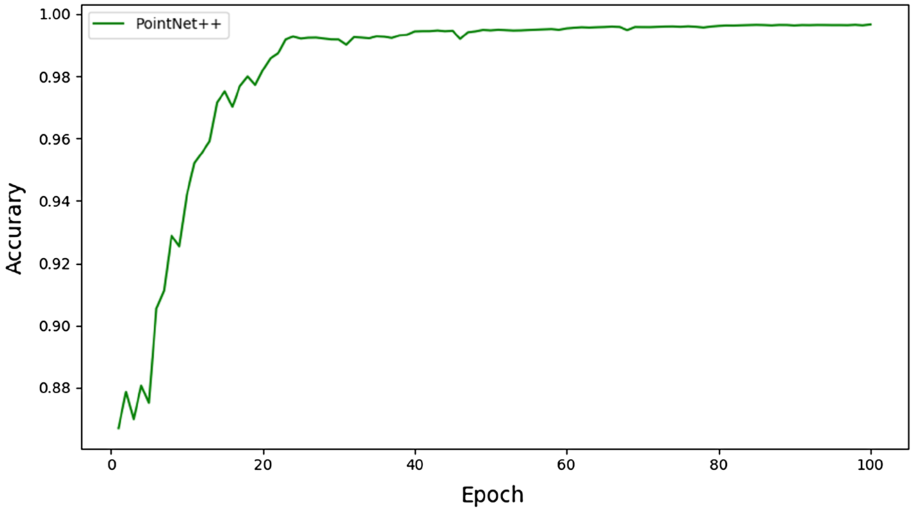
According to the results presented in Table 7, this study evaluates the performance of the two-stage scheme by calculating the mean values of the mentioned metrics on ten different plants. SegCompletion, which combines segmentation and completion within a single network, outperforms other advanced point cloud completion methods by producing higher-quality results. In contrast, the other methods, although successful in completing point clouds with regular shapes and continuous surfaces, still face challenges when dealing with objects that exhibit morphologically diverse structures and surface discontinuities, which are commonly encountered in real-world scenarios.

Generative methods, such as PF-Net and DeCo, have been capable of learning the complete structure of the input plant. However, they have often struggled to accurately place the generated points in the correct morphological positions, resulting in a mismatch with the remaining part of the input shape. In contrast, SegCompletion, which relies on segmentation and clustering networks, demonstrates the ability to differentiate between various morphological features. This enables SegCompletion to effectively segment the complex morphological structure into regular shapes and continuous surfaces. Referring to Fig. 8, it is evident that SegCompletion excels at accurately locating and completing missing regions within the point cloud of cotton plants. The completed point clouds display a uniform distribution, indicating the successful integration of the added data.

Point cloud segmentation and cluster network

The segmentation and clustering network was utilized to distinguish the different features of morphology. It successfully partitioned morphologically diverse structures or discontinuous surfaces into regular shapes and continuous surfaces. PointNet++ was employed as a network for segmenting the point cloud. The segmentation accuracy of PointNet++ gradually increased as the Epoch value increased. Once the Epoch value exceeded 10, the accuracy improvement tended to plateau, reaching a value of 98%, which satisfied the experimental requirements as illustrated in Fig. 9. In this study,

Fig. 9 Relationship between PointNet++ recognition accuracy and Epochs



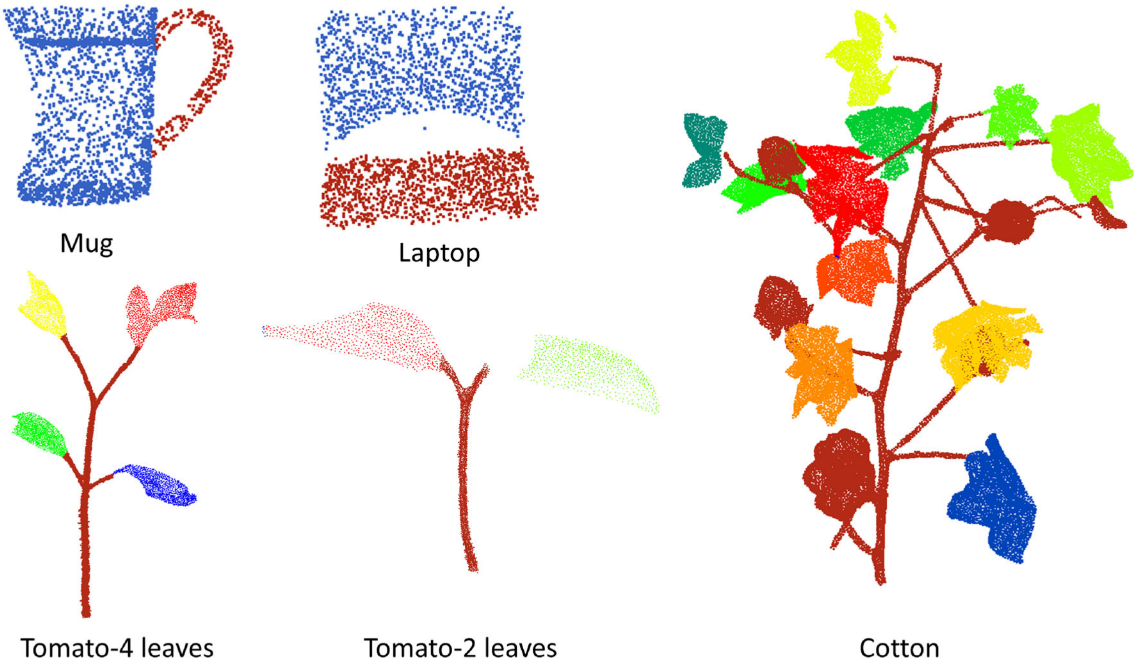


Fig. 10 Results of qualitative point cloud segmentation and clustering using point cloud incompleteness on the Pheno4D, ShapeNet, and Cotton3D datasets

an Epoch value of 100 was selected, and each iteration of the PointNet++ segmentation achieved an accuracy of over 98%. HDBSCAN was found to be effective for clustering the segmented components.

This paper presented the results of point cloud segmentation across various object categories, including Pheno4D, ShapeNet, and cotton plants in the Cotton3D dataset. For the objects in the ShapeNet dataset, the missing and complete parts were bifurcated based on their morphological characteristics. The morphology of plants was complex and diverse, and the leaves and stems were not necessarily continuous in the Pheno4D and Cotton3D datasets. Therefore, segmentation and clustering techniques were utilized to complete the point cloud representation of cotton leaves. Figure 10 demonstrates the qualitative effects of the point cloud segmentation and clustering network on both the ShapeNet and Cotton3D datasets.

The clustering algorithm's results were significantly influenced by the initial parameter values. The clustering effect was measured using the silhouette coefficient [68], where a higher value indicated a better clustering result. Through experimental validation, the maximum silhouette coefficient (Online Appendix Tables 1–4) was selected as the initial

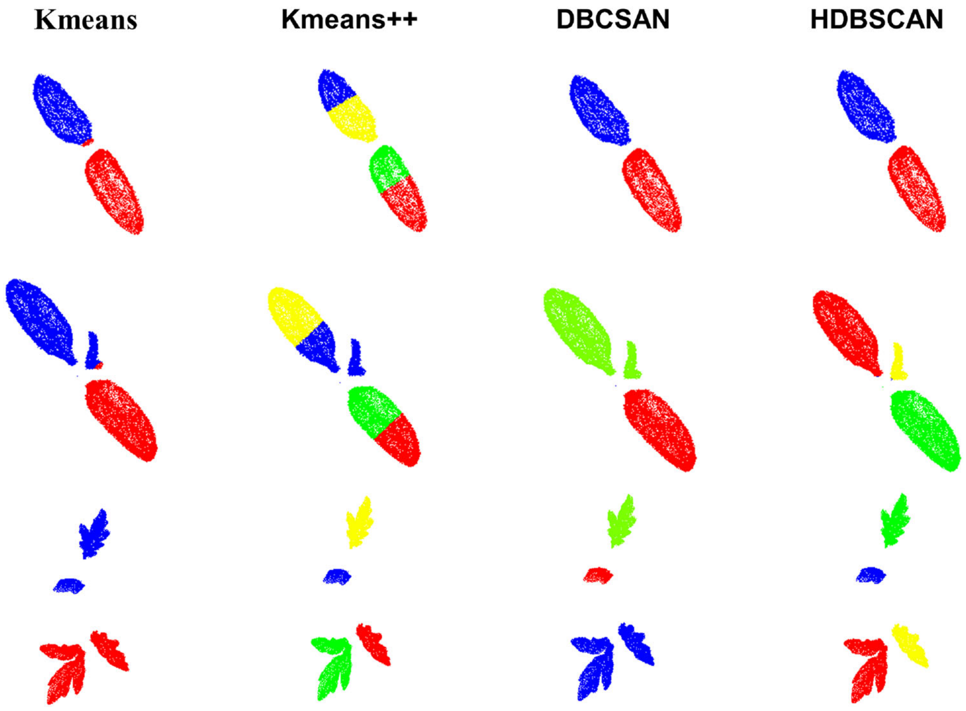


Fig. 11 Visualization of HDBSCAN in comparison to K-Means, K-Means++, and DBSCAN in the Pheno4D dataset. The number of clusters of K-Means is n_clusters = 2, the number of clusters of K-Means + + is n_clusters = 4, the ε-neighborhood of any point of DBSCAN is eps = 0.1, the minimum of MinPts points in the ε-neighborhood of DBSCAN is min_points = 400, and the minimum of any points of HDBSCAN is min_points = 400

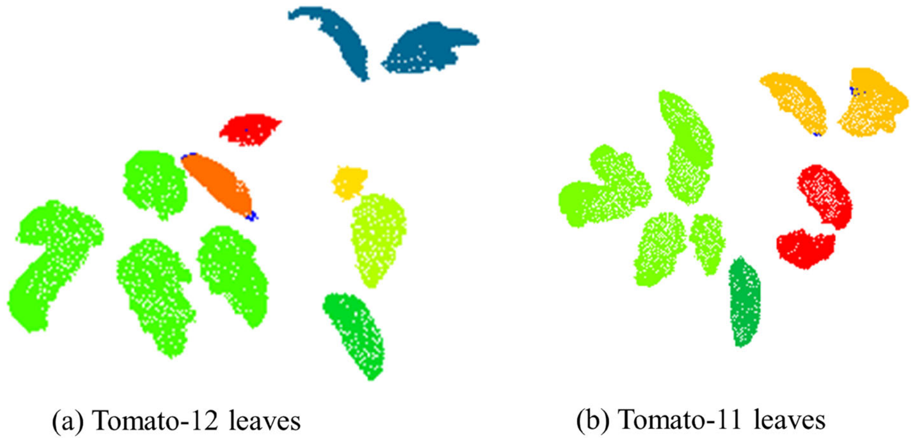


Fig. 12 Clustering results of HDBSCAN on two tomato plants. **a** This tomato plant has 12 leaves, but HDBSCAN only gathers eight, seven of which are incorrectly identified. **b** This tomato plant has eleven leaves, but HDBSCAN only gathers five, ten of which are incorrectly identified

parameter for each clustering algorithm and integrated into the Pheno4D dataset.

In different growth stages of tomatoes, HDBSCAN exhibited superior clustering performance, especially when considering varying leaf counts. In the first row of Fig. 11, representing the tomato seedling stage with only 2 leaflets, both the K-Means and K-Means++ algorithms produce erroneous clustering results. However, DBSCAN and HDBSCAN accurately identify the clusters. Transitioning to the second row, K-Means and K-Means++ still yield suboptimal clustering results, while DBSCAN fails to differentiate the intermediate leaflets. In contrast, HDBSCAN achieves better clustering performance. These results further emphasized the superiority of HDBSCAN in handling the complexity and variability of tomato plant structures throughout their growth stages. By accurately detecting and grouping the data points, HDBSCAN proved to be a robust and reliable clustering algorithm for analyzing tomato plant morphology and facilitating precise phenotypic characterization.

Limitations

The clustering results have a significant impact on the successful completion of leaves, especially when plants have multiple closely spaced leaves. In such cases, clustering algorithms, such as HDBSCAN, play a crucial role in distinguishing individual leaves from point cloud data. The effectiveness of the clustering algorithm depends on factors such as the distances between the leaves and the angles of the petioles. When the leaves are widely spread apart, algorithms such as HDBSCAN can accurately identify and complete each leaf. However, challenges arise when the leaves are closely spaced, as the clustering algorithm may struggle to differentiate them, leading to difficulties in leaf completion, as demonstrated in Fig. 12. Therefore, it is essential to consider the clustering results to achieve successful leaf completion, particularly in cases where plants have multiple closely spaced leaves.

Conclusions

The proposed network, SegCompletion, was specifically designed for point cloud completion, which involved reconstructing complete 3D shapes from partial geometries with various shape structures and noncontinuous surfaces. PointNet++ and HDBSCAN were utilized to implement morphological segmentation before point cloud completion, allowing for effective segmentation of point cloud instances belonging to the same object category. The multiscale generative network facilitated advanced patching of missing point clouds based on shared geometric features derived from feature points. Additionally, a straightforward and efficient uniform loss function was introduced to minimize the variance in average distances between patch centers and their corresponding closest neighbors. Extensive experiments were conducted on ShapeNet, Pheno4D, and our self-collected Cotton3D dataset to evaluate the effectiveness of the proposed method. The results demonstrated the superiority of the SegCompletion method over other approaches discussed in the literature, establishing its prominence in point cloud completion tasks.

This paper serves as an initial step toward achieving high-quality completion of single object point clouds, which is a typical challenge in the field. An important direction for future research is to extend the completion framework to handle multi-object scenes. Currently, SegCompletion focuses on generating the complete geometric representation of individual objects. However, the goal is to segment and complete individual objects within larger scenes, such as tables, chairs, or cars in indoor and outdoor environments like ScanNet and KITTI datasets. Additionally, we believe that incorporating appropriate joint optimization strategies and enabling end-to-end training across all hierarchical levels will further enhance the performance of our method. Despite these considerations, this research makes a significant contribution to the advancement of 3D shape completion and understanding. It provides valuable insights and methodologies with potential applications across various domains, benefiting industries, academia, and society.

Author contributions C-JS designed and managed the project. Z-BY performed data analyses. M-DSh, L-PC, and C-JS wrote the manuscript. RN organized charts and tables. NY collected ShapeNet data. Z-QF, S-QSh and F-YL collected the Cotton3D data. All authors revised and approved the final manuscript.

Funding This work was supported by “the National Natural Science Foundation of China (61961035)”, “the First-class Undergraduate Programmers Foundation in Computer Graphics at Tarim University (TDYLKC202231)”, and “the First-class Major in the Internet of Things Engineering at Tarim University (22/22000030126)”.

Data availability Data from this study are not publicly available.

Declarations

Conflict of interest The authors declare no conflict of interest.

Institutional review board statement Not applicable.

Informed consent Not applicable.

References

1. Diaz N, Gallo O, Caceres J, Porras H (2021) Real-time ground filtering algorithm of cloud points acquired using terrestrial laser scanner (TLS). Int J Appl Earth Obs Geoinf 105:102629
2. Lin C, Hu F, Peng J, Wang J, Zhai R (2022) Segmentation and stratification methods of field maize terrestrial LiDAR point cloud. Agriculture 12(9):1450
3. Ma X, Li X, Song J (2022) Point cloud completion network applied to vehicle data. Sensors 22(19):7346
4. González-Barbosa J-J, Ramírez-Pedraza A, Ornelas-Rodríguez F-J, Cordova-Esparza D-M, González-Barbosa E-A (2022) Dynamic measurement of portos tomato seedling growth using the Kinect 2.0 sensor. Agriculture 12(4):449
5. Moreno H, Bengochea-Guevara J, Ribeiro A, Andújar D (2022) 3D assessment of vine training systems derived from ground-based RGB-D imagery. Agriculture 12(6):798
6. Lu H, Shi H (2020) Deep learning for 3D point cloud understanding: a survey. arXiv Preprint arXiv:2009.08920.
7. Engel J, Schops T, Cremers D (2014) In LSD-SLAM: large-scale direct monocular SLAM, computer vision-ECCV 2014: 13th European conference. Springer International Publishing, Zurich, pp 834–849
8. Mur-Artal R, Montiel J, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 31(5):1147–1163
9. Du R, Ma Z, Xie P, He Y, Cen H (2023) PST: plant segmentation transformer for 3D point clouds of rapeseed plants at the podding stage. ISPRS J Photogramm Remote Sens 195:380–392

10. Han Z, Wang X, Vong C-M, Liu Y-S, Zwicker M, Chen C (2019) 3DViewGraph: learning global features for 3D shapes from a graph of unordered views with attention. arXiv Preprint arXiv:1905.07503
11. Han Z, Liu X, Liu Y-S, Zwicker M (2019) Parts4Feature: learning 3D global features from generally semantic parts in multiple views. arXiv Preprint arXiv:1905.07506
12. Wen X, Han Z, Liu X, Liu Y-S (2020) Point2SpatialCapsule: aggregating features and spatial relationships of local regions on point clouds using spatial-aware capsules. IEEE Trans Image Process 29:8855–8869
13. Liu X, Han Z, Hong F, Liu Y-S, Zwicker M (2020) LRC-Net: learning discriminative features on point clouds by encoding local region contexts. Comput Aided Geom Des 79:101859
14. Yuan W, Khot T, Held D, Mertz C, Hebert M (2018) In PCN: Point Completion Network, 2018 international conference on 3D vision (3DV), August 01, 2018. IEEE, pp 728–737
15. Yang Y, Feng C, Shen Y, Tian D (2017) FoldingNet: point cloud auto-encoder via deep grid deformation. In: IEEE conference on computer vision and pattern recognition, June 18–22, 2018. pp 206–215
16. Wang X, Ang MH Jr, Lee GH (2020) Cascaded refinement network for point cloud completion with self-supervision. IEEE Trans Pattern Anal Mach Intell 44(11):8139–8150
17. Dai A, Ruizhongtai Qi C, Nießner M (2017) Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In: IEEE conference on computer vision and pattern recognition, July 21–26, 2017. pp. 5868–5877
18. Gao J, Chen W, Xiang T, Fuji Tsang C, Jacobson A, McGuire M, Fidler S (2020) Learning deformable tetrahedral meshes for 3D reconstruction. Adv Neural Inf Process Syst 33:9936–9947
19. Deng Y, Yang J, Tong X (2021) Deformed implicit field: modeling 3D shapes with learned dense correspondence. In: IEEE/CVF conference on computer vision and pattern recognition, June 20–25, 2021. pp 10286–10296
20. Wen X, Xiang P, Han Z, Cao Y-P, Wan P, Zheng W, Liu Y-S (2021) PMP-Net: point cloud completion by learning multi-step point moving paths. In: IEEE/CVF conference on computer vision and pattern recognition, June 20–25, 2021. pp 7443–7452
21. Yin K, Huang H, Cohen-Or D, Zhang H (2018) P2P-NET: bidirectional point displacement net for shape transform. ACM Trans Graph (TOG) 37(4):1–13
22. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space. Adv Neural Inf Process Syst 30:1–10
23. Achlioptas P, Diamanti O, Mitliagkas I, Guibas L (2017) Learning representations and generative models for 3D point clouds. In: International conference on machine learning. PMLR. pp 40–49
24. Huang Z, Yu Y, Xu J, Ni F, Le X (2020) PF-Net: point fractal network for 3D point cloud completion. In: IEEE/CVF conference on computer vision and pattern recognition, June 13–19, 2020. pp 7662–7670
25. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D ShapeNets: a deep representation for volumetric shapes. In: IEEE conference on computer vision and pattern recognition, June 7–12, 2015. pp 1912–1920
26. Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H, Xiao J, Yi L, Yu F (2015) ShapeNet: an information-rich 3D model repository. arXiv Preprint, arXiv:1512.03012
27. Pan L, Chen X, Cai Z, Zhang J, Zhao H, Yi S, Liu Z (2021) Variational relational point completion network. In: IEEE/CVF conference on computer vision and pattern recognition, June 20–25, 2021. pp 8524–8533
28. Campello RJGB, Moulavi D, Sander J (2013) Density-based clustering based on hierarchical density estimates. Knowl Inf Syst 42(3):1–29
29. Wen X, Li T, Han Z, Liu Y-S (2020) Point cloud completion by skip-attention network with hierarchical folding. In: IEEE/CVF conference on computer vision and pattern recognition, June 13–19, 2020. pp 1939–1948
30. Yuan J, Chen C, Yang W, Liu M, Xia J, Liu S (2020) A survey of visual analytics techniques for machine learning. Comput Vis Media 7:3–36
31. Wen X, Han Z, Cao Y-P, Wan P, Zheng W, Liu Y-S (2021) Cycle4Completion: unpaired point cloud completion using cycle transformation with missing region coding. In: IEEE/CVF conference on computer vision and pattern recognition, March 01, 2021. pp 13080–13089
32. Wen X, Xiang P, Han Z, Cao YP, Wan P, Zheng W, Liu YS (2023) PMP-Net++: point cloud completion by transformer-enhanced multi-step point moving paths. IEEE Trans Pattern Anal Mach Intell 45(1):852–867
33. Berger M, Tagliasacchi A, Seversky L, Alliez P, Levine J, Sharf A, Silva C (2014) State of the art in surface reconstruction from point clouds. Eurograph 2014 State Art Rep 1(1):161–185
34. Shao T, Xu W, Zhou K, Wang J, Li D, Guo B (2012) An interactive approach to semantic modeling of indoor scenes with an RGBD camera. ACM Trans Graph 31(6):1–11
35. Nguyen DT, Hua B-S, Tran M-K, Pham Q-H, Yeung S-K (2016) A field model for repairing 3D shapes. In: IEEE conference on computer vision and pattern recognition, June 27–30, 2016. pp 5676–5684
36. Fu Z, Hu W, Guo Z (2019) Local frequency interpretation and non-local self-similarity on graph for point cloud in painting. IEEE Trans Image Process 28(8):4087–4100
37. Sung M, Kim VG, Angst R, Guibas L (2015) Data-driven structural priors for shape completion. ACM Trans Graph 34(6):1–11
38. Kalogerakis E, Chaudhuri S, Koller D, Koltun V (2012) A probabilistic model for component-based shape synthesis. ACM Trans Graph 31(4):1–11
39. Martinovic A, Gool LV (2013) Bayesian grammar learning for inverse procedural modeling. In: IEEE conference on computer vision and pattern recognition, June 23–28, 2013. pp 201–208
40. Shen CH, Fu H, Chen K, Hu SM (2012) Structure recovery by the part assembly. ACM Trans Graph 31(6):1–11
41. Zhang J, Chen X, Cai Z, Pan L (2021) Unsupervised 3D shape completion through GAN inversion. In: IEEE/CVF conference on computer vision and pattern recognition, June 20–25, 2021. pp 1768–1777
42. Sarmad M, Lee HJ, Kim YM (2019) In RL-GAN-Net: a reinforcement learning agent controlled GAN network for real-time point cloud shape completion, the IEEE/CVF conference on computer vision and pattern recognition, April 01, 2019. IEEE Computer Society, pp 5898–5907
43. Hu T, Han Z, Zwicker M (2019) 3D shape completion with multi-view consistent inference. In: AAAI conference on artificial intelligence, vol 34, Jan 27–Feb 1, 2019. pp 10997–11004
44. Fei B, Yang W, Chen W-M, Li Z, Li Y, Ma T, Hu X, Ma L (2022) Comprehensive review of deep learning-based 3D point cloud completion processing and analysis. IEEE Trans Intell Transp Syst 23(12):22862–22883
45. Zhang J, Zhao X, Chen Z, Lu Z (2019) A review of deep learning-based semantic segmentation for point cloud. IEEE Access 7:179118–179133
46. Mao Y, Sun X, Diao W, Chen K, Guo Z, Lu X, Fu K (2022) Semantic segmentation for point cloud scenes via dilated graph feature aggregation and pyramid decoders. arXiv Preprint arXiv:2204.04944

47. Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM (2019) Dynamic graph cnn for learning on point clouds. ACM Trans Graph (tog) 38(5):1–12
48. Thomas H, Qi CR, Deschaud J-E, Marcotegui B, Goulette F, Guibas LJ (2019) In Kpconv: flexible and deformable convolution for point clouds. In: IEEE/CVF international conference on computer vision, Oct 27–Nov 2, 2019. pp 6411–6420
49. Nam J, Nam T-J (2017) In TransPoint: real-time remote lecturing via adaptive transparency. In: 2017 conference on designing interactive systems, June 10–14, 2017. pp 631–635
50. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3d shapenets: a deep representation for volumetric shapes. In: IEEE conference on computer vision and pattern recognition, June 7–12, 2015. pp 1912–1920
51. Qi CR, Su H, Mo K, Guibas LJ (2016) PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE conference on computer vision and pattern recognition, June 27–30, 2016. pp 652–660
52. Bajal E, Bhatia M, Hooda M, Katara V (2022) A review of clustering algorithms: comparison of DBSCAN and K-mean with oversampling and t-SNE. Recent Patents Eng 16(2):17–31
53. Lloyd S (1957) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137
54. Johnson SC (1967) Hierarchical clustering schemes. Psychometrika 32(3):241–254
55. Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416
56. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
57. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Generative image inpainting with contextual attention. In: IEEE conference on computer vision and pattern recognition, June 18–23, 2018. pp 5505–5514
58. Tan J, Jing L, Huo Y, Tian Y, Akin O (2019) LGAN: lung segmentation in CT scans using generative adversarial network. Comput Med Imaging Graph 87:101817
59. Tian Z, Shen C, Chen H, He T (2019) FCOS: fully convolutional one-stage object detection. In: IEEE/CVF international conference on computer vision, Oct 27–Nov 2, 2019. pp 9627–9636
60. Zhang J, Chen X, Cai Z, Pan L, Zhao H, Yi S, Kiat Yeo C, Dai B, Change Loy C (2021) Unsupervised 3D shape completion through GAN inversion. arXiv Preprint arXiv:2104.13366.
61. Fan H, Su H, Guibas L (2017) A point set generation network for 3D object reconstruction from a single image. In: IEEE conference on computer vision and pattern recognition, July 21–26, 2017. pp 605–613
62. Liu M, Sheng L, Yang S, Shao J, Hu S-M (2019) Morphing and sampling network for dense point cloud completion. In: AAAI conference on artificial intelligence, vol 34, Jan 27–Feb 1. pp 11596–11603
63. Li R, Li X, Fu C-W, Cohen-Or D, Heng P-A (2019) In PU-GAN: a point cloud upsampling adversarial network. In: IEEE/CVF international conference on computer vision, July 01, 2019. pp 7203–7212
64. Schunck D, Magistri F, Rosu RA, Cornelissen A, Chebrolu N, Paulus S, Leon J, Behnke S, Stachniss C, Kuhlmann H, Klingbeil L (2021) Pheno4D: a spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. PLoS One 16(8):e0256340
65. Xie H, Yao H, Zhou S, Mao J, Zhang S, Sun W (2020) In GRNet: gridding residual network for dense point cloud completion, computer vision-ECCV 2020: 16th European conference, June 01, 2020; Springer International Publishing, Cham, Glasgow, pp 365–381
66. Alliegro A, Valsesia D, Fracastoro G, Magli E, Tommasi T (2021) Denoise and contrast for category agnostic shape completion. arXiv Preprint arXiv:2103.16671
67. Xiang P, Wen X, Liu Y-S, Cao Y-P, Wan P, Zheng W, Han Z (2021) In SnowflakeNet: point cloud completion by snowflake point deconvolution with skip-transformer, the IEEE/CVF international conference on computer vision, August 01, 2021. pp 5499–5509
68. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20(1):53–65