

Article

Completing 3D Point Clouds of Thin Corn Leaves for Phenotyping Using 3D Gridding Convolutional Neural Networks

Ying Zhang ^{1,2}, Wei Su ^{1,2,*} , Wancheng Tao ^{1,2}, Ziqian Li ^{1,2}, Xianda Huang ^{1,2} , Ziyue Zhang ^{1,2} and Caisen Xiong ^{1,2}

¹ College of Land Science and Technology, China Agricultural University, Beijing 100083, China

² Key Laboratory of Remote Sensing for Agri-Hazards, Ministry of Agriculture, Beijing 100083, China

* Correspondence: suwei@cau.edu.cn

Abstract: Estimating the complete 3D points of crop plants from incomplete points is vital for phenotyping and smart agriculture management. Compared with the completion of regular man-made objects such as airplanes, chairs, and desks, the completion of corn plant points is more difficult for thin, curled, and irregular corn leaves. This study focuses on MGRNet+OA, which is based on GRNet, to complete a 3D point cloud of thin corn plants. The developed MGRNet+OA was accompanied by gridding, multi-scale **3DCNN**, gridding reverse, cubic feature sampling, and offset-attention. In this paper, we propose the introduction of a 3D grid as an intermediate representation to regularize the unorganized point cloud, use multi-scale predictive fusion to utilize global information at different scales, and model the geometric features by adding offset-attention to compute the point position offsets. These techniques enable the network to exhibit good adaptability and robustness in dealing with irregular and varying point cloud structures. The accuracy assessment results show that the accuracy of completion using **MGRNet+OA** is superlative, with a CD ($\times 10^{-4}$) of 1.258 and an *F-Score@1%* of 0.843. MGRNet+OA is the most effective when compared with other networks (**PCN**, **shape inversion**, the original **GRNet**, **SeedFormer**, and **PMP-Net++**), and it improves the accuracy of the CD ($\times 10^{-4}$)/*F-Score@1%* with $-15.882/0.404$, $-15.96/0.450$, $-0.181/0.018$, $-1.852/0.274$, and $-1.471/0.203$, respectively. These results reveal that the developed MGRNet+OA can be used to complete a 3D point cloud of thin corn leaves for phenotyping.



Citation: Zhang, Y.; Su, W.; Tao, W.; Li, Z.; Huang, X.; Zhang, Z.; Xiong, C. Completing 3D Point Clouds of Thin Corn Leaves for Phenotyping Using 3D Gridding Convolutional Neural Networks. *Remote Sens.* **2023**, *15*, 5289. <https://doi.org/10.3390/rs15225289>

Academic Editor: Mario Cunha

Received: 7 September 2023

Revised: 9 October 2023

Accepted: 1 November 2023

Published: 9 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

LiDAR is an emerging active remote sensing technology that can acquire the 3D structural information of objects directly, quickly, and accurately [1]. Terrestrial laser scanning (TLS) is a highly accurate 3D stereoscopic measurement method with high accuracy, which can collect the points returning to crop plants automatically [2,3], accurately, and efficiently, providing more possibilities for crop phenotyping [4]. Terrestrial laser scanning (TLS) provides more possibilities for crop phenotyping with its high resolution, high accuracy, and high efficiency [5]. Unfortunately, there is one issue that should be pointed out: there is unavoidable occlusion after the crop canopy is closed when the crop plants and leaves are overlapped. And this occlusion will result in the incompleteness of LiDAR points returning to crop plants, which will lead to the inaccurate estimation of phenotyping traits. Therefore, the completion of LiDAR points in crop plants is vital for phenotyping. Corn (*Zea mays*) is a C4 crop that contributes significantly to world food security by providing both food and fodder. Corn leaves are thin and curly, and their 3D completion for phenotyping is crucial but difficult. So, we explored the completion of corn leaves' LiDAR points in this study.

The traditional approaches to 3D completion can be grouped as 3D-model-based [6] and 3D-geometry-based methods. For the 3D-model-based methods, completion is achieved via fitting the similarity of a 3D structure between the target object and its 3D model projection [7]. And the prior 3D structural characteristics of the target object are used for modeling description, which is more suitable for completing objects with regular shapes [8–10]. Unfortunately, the generalization of the 3D-model-based method is poor [11,12]. For the 3D-geometry-based methods [13], occlusions are completed via the 3D indices, including shading [14–17], occlusion [18], texture [19–21], and multi-visual chrematistics [22]. Therefore, the 3D-geometry-based method offers great generalization without targeting any specific class of objects [23]. However, only the existing visible parts of the 3D structure can be completed using the 3D-geometry-based method, and the missing part cannot be completed. Therefore, these two kinds of approaches are not applicable to completing the points of thin corn leaves.

With the rapid development of artificial intelligence techniques, deep learning has shown great potential in the completion of 3D objects. These methods can be divided into 2D-image-based methods, point-based methods, voxel-based methods, and mesh-based methods, depending on the type of input data. For 2D-image-based methods, multi-view 2D images are used to recover the depth images of objects using 2D convolutional neural networks with great efficiency [24–26]. However, these methods are more sensitive to internal leaf occlusion, and 3D structural information cannot be acquired. For voxel-based approaches, generalized voxels are used as the intermediate media, extending convolutional neural networks from 2D to 3D for completion [27–31]. The main difficulties of these methods lie in the huge computation and the loss of points' precision during the point cloud voxelization process. Therefore, some studies have explored the multilayer perceptron (MLP) to process point clouds directly, using maximum pooling to aggregate point-to-point information globally or hierarchically [32,33]. However, this approach fails to consider the connectivity between points and the contextual information of neighboring points fully during the task of shape completion. Fortunately, some studies have explored extracting data features from polygonal meshes using convolutional neural networks, aiming to acquire the correct geometry of the target object and complete the 3D target objects [34,35]. However, this method is sensitive to the density of points, and the coordinates of points cannot be changed before and after convolution. So, the convolutional neural network is not applicable to point cloud completion [36]. GRNet has introduced a 3D grid as an intermediate representation of regularized disordered point clouds to obtain the geometric structure and contextual information of the target object point clouds [37]. Xie completed the missing points of man-made objects including airplanes, chairs, and lamps accurately. Recent studies have improved the accuracy of network predictions by introducing various processing mechanisms. PMP-Net++ incorporates the point-wise mutual promotion mechanism, which provides an approach to the interaction between global-local information to improve the accuracy of point cloud completion [38]. SeedFormer is a method based on the Transformer architecture, combining self-attention and fully connected layers to capture information from local and global contexts in point clouds [39]. Additionally, ProxyFormer provides a new perspective by using proxy points and a self-attention mechanism to process point cloud data to accomplish the task of point cloud completion [40]. Some studies have started to introduce other features to predict the point cloud complementation task; e.g., Federico proposed a single RGB-D frame to predict the 3D shapes of a complete fruit during operation [41]. Unfortunately, there have been no experiments on the potential of the networks in completing thin and irregular vegetation leaves. The complex and irregular feature structure of corn leaves presents some challenges for the completion task. So, we explored the potential of completing thin corn leaves' points in this study.

Despite the rapid development of LiDAR and deep learning technologies, there still are challenges in point cloud completion. Compared with the completion of regular man-made objects, such as airplanes, chairs, and lamps, the difficulties and challenges of completing thin, curled corn leaves' LiDAR points are as follows:

- (1) There is no available experimental dataset of a corn plant point cloud used for the training and validation of completion. The publicly shared datasets mainly represent airplanes, chairs, table lamps, etc. There is no available plant point cloud dataset for completion currently.
- (2) The irregular structure of thin, curled corn leaves also poses a challenge. Compared with man-made objects, corn leaves are irregular, thin, curled, and varied in different growing seasons. This results in more difficulties in developing a gridding residual network for completion.

Given these difficulties and challenges, we developed a novel model called MSGRNet+OA, based on the GRNet architecture. To improve the training performance of the network and the adaptation of the network to complete the 3D point cloud of corn plants, the network layers were deepened. An offset-attention module was introduced in the upper layer to capture more features of the corn shape. The introduction of a multi-scale fusion module into the network increased the richness of information, preserved structural details, improved computational efficiency, and increased the robustness of the algorithm. Then, we explored the potential of MSGRNet+OA in completing the 3D point cloud of thin corn plants. The objectives of this study focused on (1) the generation of a training and testing dataset for corn leaves' completion, (2) the modification of the network structure, the investigation of the sensitivity of different network parameters in the task of corn leaves' completion, and the training and testing of the network for corn leaves' completion, and (3) experiments on the completion of corn leaves' points during different growth periods.

2. Methodology

The network was based on the gridding residual network (GRNet), which is an approach to completing the missing points in 3D space. In order to capture the 3D structural and contextual features of objects, a 3D grid was introduced as an intermediate representation to regularize the disorganized points. Firstly, the original corn LiDAR point cloud is converted to 3D grids, which act as inputs for the network. Secondly, a 3D convolutional neural network is built to complete the residual parts of the 3D grid [42], and gridding reverse converts the 3D output grid to a coarse point cloud. Thirdly, cubic feature sampling is performed to extract features for each point in the coarse point cloud. Finally, the coarse point cloud and point features are input into an MLP to complete the LiDAR point cloud. Therefore, GRNet is suitable for completing the 3D point cloud of thin corn leaves. However, the completion of corn plant points is more difficult for thin, curled, and irregular corn leaves; we diligently scrutinized the network architecture to discover some potential scopes for improvement.

The irregular structure of corn leaves and the structural changes at different growth stages make it necessary for the network to consider and account for these problems when completing the task. In response to this problem, we introduced multi-scale prediction fusion and integrated the mechanism of offset-attention. First, multi-scale fusion is of great significance in the completion of corn point clouds. It can increase information richness, preserve structural details, improve computational efficiency, and increase the robustness of the algorithm [43]. Moreover, it is difficult to determine the importance of each point feature in each layer of GRNet. In order to extract important information from corn, handle irregular structures, and improve computational efficiency, an attention mechanism was introduced to increase the expressiveness of the network and the accuracy of the task to improve the results of corn point cloud completion. Meanwhile, to obtain the data features of the tiny parts of corn, the number of 3D CNN layers was increased. The architecture of the network is described in Section 2.1, and the loss function of the network is introduced in Section 3.2.

2.1. Modification and Architecture of the Gridding Residual Network

There are six steps for corn leaves' point completion. (1) Gridding is performed to convert LiDAR points into a 3D grid with $G = \langle V, W \rangle$, given the incomplete point cloud P

as an input, where V is the vertex set of G and W . (2) A multi-scale 3D CNN is built and deepened to train the objects' traits and complete the 3D grid. (3) The completed 3D grids are converted into coarse point clouds. (4) The features of each point in the coarse point cloud are extracted. (5) The point cloud is completed by inputting the coarse point cloud and the extracted feature into MLP. (6) An offset-attention module is added to improve the network performance. This architecture is depicted in Figure 1.

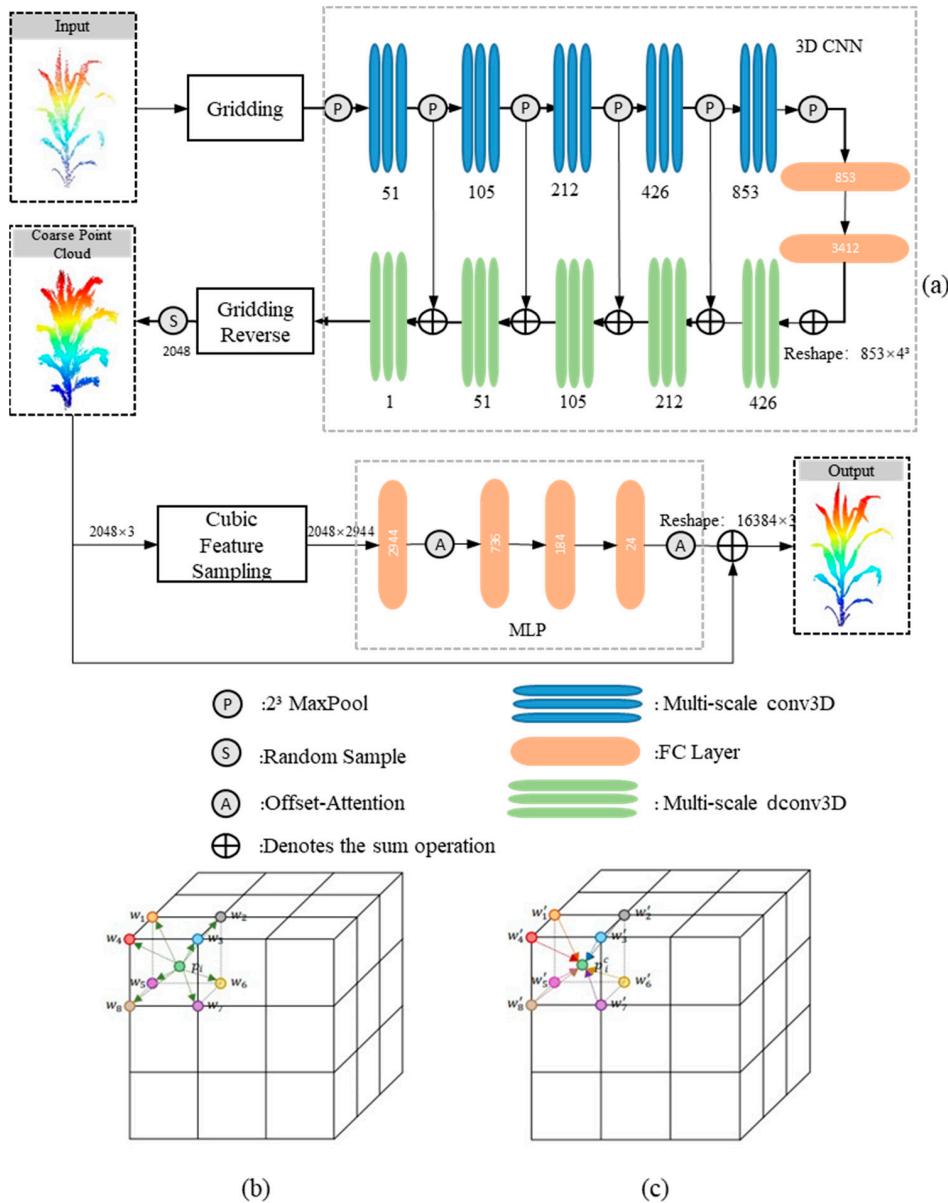


Figure 1. Architecture of the multi-scale gridding residual network: (a) the main structure of the network, including gridding, the multi-scale 3D CNN, gridding reverse, cubic feature sampling, MLP, and the attention module; (b,c) diagrams of gridding and gridding reverse, respectively.

2.1.1. The Gridding of Processed LiDAR Points

The gridding process introduces a 3D grid as an intermediate representation to regularize the unorganized point cloud, and the differentiable grid layer was proposed to convert the disordered and unorganized point cloud $P = \{p_i\}_{i=1}^n$ into a regular 3D grid $G = \langle V, W \rangle$ while preserving the spatial structure of the point cloud with $p_i \in \mathbb{R}^3$, $V = \{v_i\}_{i=1}^{N^3}$, $W = \{w_i\}_{i=1}^{N^3}$, $v_i \in \left\{ \left(-\frac{N}{2}, -\frac{N}{2}, -\frac{N}{2} \right), \dots, \left(\frac{N}{2} - 1, \frac{N}{2} - 1, \frac{N}{2} - 1 \right) \right\}$, $w_i \in \mathbb{R}$; n is the num-

ber of points in the point cloud P, and N is the resolution of the 3D grid G. A cell is defined as a 3D cube with eight vertices. For each vertex, the weight of each grid vertex is computed using the distance weightings (Figure 1). The weightings for each 3D grid are calculated as follows:

$$w_i = \sum_{p \in N(v_i)} \frac{w(v_i, p)}{|N(v_i)|} \quad (1)$$

where $|N(v_i)|$ is the number of neighboring points of vertex i , and $w(v_i, p)$ is calculated as follows:

$$w(v_i, p) = (1 - |x_i^v - x|)(1 - |y_i^v - y|)(1 - |z_i^v - z|) \quad (2)$$

where, $w(v_i, p)$ is the interpolation function, $v_i = (x_i^v, y_i^v, z_i^v)$ is the vertices of the 3D grid cell, and $p = (x, y, z)$ is defined as a neighboring point of vertex v_i .

2.1.2. Construction of a Multi-Scale 3D CNN

The original GRNet uses a 3D CNN with skip connections [44] to complete the missing parts of the point cloud. Three-dimensional CNN [42] follows the idea of U-Net [45] connections and proposes the 3D encoder-decoder. Given W as the input, the 3D CNN is defined as $W' = 3DCNN(W)$, where $W' = \{w'_i\}_{i=1}^{N^3}$, $w'_i \in \mathbb{R}$. The encoder of the 3D CNN mainly consists of four 3D convolutional layers; each layer of the original GRNet includes 4^3 filters with step 2, followed by BN, Leaky ReLU, and a maximum pooling layer with a kernel size of 2^3 . With the development of backbone CNNs, some studies have revealed that the strong representation ability at multiple scales can bring some improvement in performance. Multi-scale fusion is a technique that combines information from different scales. It involves merging data or features from multiple scales to obtain a more comprehensive and accurate representation. The purpose of multi-scale fusion is to improve network performance by combining information from different scales. This includes improving the richness of information in the data, preserving details and structures, increasing robustness, and improving computational efficiency. First, the multiscale fusion approach allows the combination of features from different scales, increasing the information richness of point cloud data and making the completion results more accurate and complete. Then, the structural details of corn plants are critical to the task of completion, such as the shape of leaves and the positions of ears. Multi-scale fusion allows structural details to be preserved at different scales, enabling a better prediction of the shape and structure of corn during the completion process. In addition, multi-scale fusion can increase computational efficiency by splitting point cloud data into different scales for separate processing. Multi-scale fusion can increase the robustness of the algorithm by integrating information from multiple scales to better predict the complex and irregular structural features of corn leaves.

Following the approach of the Inception network, the simplest way to augment the network with multi-resolution analysis capability is to parallelize the convolution operations of different scales. Despite the gain in performance, the introduction of additional parallel convolutional layers increases memory requirements excessively. Therefore, we used a series of smaller and lighter-weight 3^3 convolution blocks to decompose larger 5^3 and 7^3 convolution layers, as shown in Figure 2. Consequently, we obtained outputs from three convolution blocks and concatenated them together to extract spatial features at different scales.

To improve the network performance and obtain more core data features, the number of 3D CNN layers was increased from the original four to five. The output channels of the convolutional layers were 32, 64, 128, 256, and 512, respectively. And the sizes of the following two fully connected layers were 2048 and 4096. The decoder consisted of four transposed convolutional layers, each of them with 4^3 filters by the step size of 2. The activation functions used were the BN function and the ReLU function.

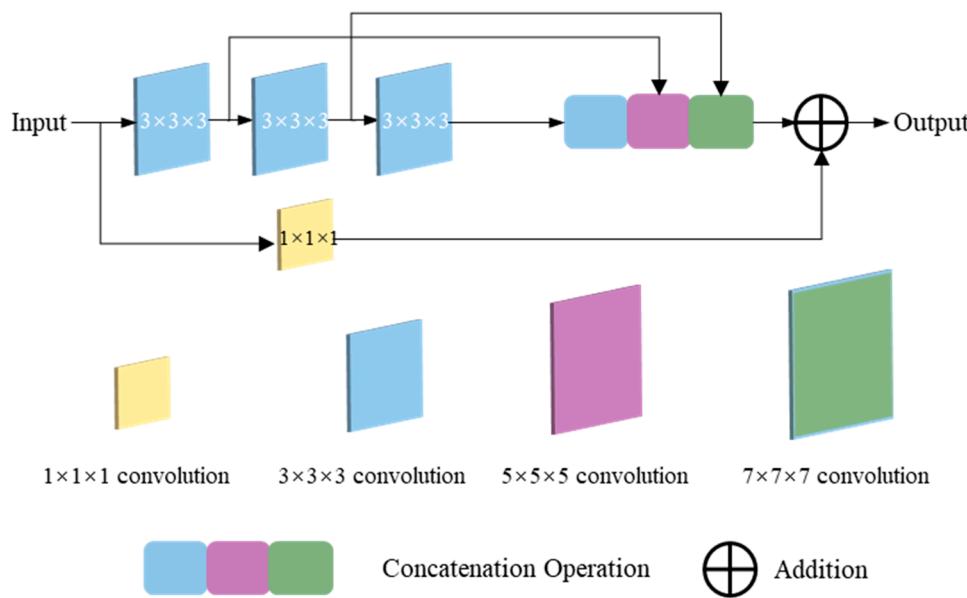


Figure 2. The structure of a multi-scale block. We used a series of 33 smaller and lighter-weight convolution blocks to decompose larger 53 and 73 convolution layers and added a residual connection (along with 1×1 filters to conserve dimensions).

2.1.3. Gridding Reverse

The gridding reverse of GRNet generates a coarse point cloud $P^c = \{p_i^c\}_{i=1}^m$ from a 3D grid $G' = \langle V, W' \rangle$, where $p_i^c \in \mathbb{R}^3$, and m is the number of points in the coarse point cloud (Figure 1). And p_i^c is determined by calculating the values and the coordinates of the eight vertices in each 3D grid, which can be expressed as follows:

$$p_i^c = \frac{\sum_{\theta \in \Theta^i} w'_\theta v_\theta}{\sum_{\theta \in \Theta^i} w'_\theta} \quad (3)$$

where, $\{v_\theta | \theta \in \Theta^i\}$ are the values of the eight vertices in the cell, $\{w'_\theta | \theta \in \Theta^i\}$ is the vertex dataset of eight vertices, and $\Theta^i = \{\theta_j^i\}_{j=1}^8$ is the index of the vertex in the i -th 3D grid cell. And the gridding reverse process will not produce p_i^c when $\sum_{\theta \in \Theta^i} w'_\theta = 0$.

2.1.4. Extracting Features and Completing via MLP

In order to acquire the contextual traits of the point cloud, GRNet uses cubic feature sampling to extract the feature $F^c = \{f_i^c\}_{i=1}^m$ from the coarse point cloud P^c , which helps the multilayer perceptron reconstruct the details of the objects. For any point p_i^c in the coarse point cloud P^c , its characteristic f_i^c is calculated via the following equation:

$$f_i^c = [f_{\theta_1^i}^v, f_{\theta_2^i}^v, \dots, f_{\theta_8^i}^v] \quad (4)$$

where [...] is the concatenation operation, and $\left\{f_{\theta_j^i}^v\right\}_{j=1}^8$ are the features of eight vertices of the 3D grid cell where p_i^c is located. In GRNet, cubic feature sampling was applied to the first three transposed convolutional layers of the 3D CNN. To reduce the redundancy of feature data, 2048 points were selected randomly from the coarse point cloud P^c , and a feature map with the size of $2048 \times (128 + 64 + 32) \times 8$ was generated.

After feature extraction, MLP was proposed to calculate and study the deviation between the coarse point cloud and the completed point cloud to recover the details of

the objects. The multilayer perceptron took the coarse point cloud P^c and its feature F^c as inputs, and the complemented point cloud $P^f = \{p_i^f\}_{i=1}^k$ was calculated as follows:

$$P^f = MLP(F^c) + Tile(P^c, r) \quad (5)$$

where $p_i^f \in \mathbb{R}^3$, k denotes the number of points in P^f , and r is the number of repetitions of the tile. In GRNet, the number of points in the coarse point cloud was 2048, and the output was 16,384, so r was set to 8. The eight offset vectors of P^c were estimated via the tile, which was the offset of the final point cloud relative to the points in the coarse point cloud. There were four fully connected layers with sizes of 1792, 448, 112, and 24 in MLP, respectively. And the output was $16,384 \times 3$ for 16,384 points.

2.1.5. Attention Mechanism

The attention mechanism plays a vital role in a deep learning task [46], which is a way to mimic human visual and cognitive systems. Introducing an attention mechanism allowed the network to automatically learn and selectively focus on important information in the input, improving the performance and generalization of the model. Corn point cloud data may contain a significant amount of redundant and irrelevant information, while the attention mechanism can assist the network in focusing on regions and features that are of critical importance. By calculating attention weights, the network can more effectively extract and focus on the key structures, shapes, and details of corn plants, improving the accuracy of the completion task. The attention mechanism can dynamically adjust the network focus and weight assignment, based on various features and contextual information in the data, helping to handle the irregular structures in corn's point cloud data.

Offset-attention is an effective 3D attention module [47] that is illustrated in Figure 3. Offset-attention was proposed to estimate the offsets between the input and attention features, which were calculated from a self-attention structure. Due to the use of point position offsets to model geometric features, offset-attention shows good adaptability and robustness in dealing with irregular and varying point cloud structures. By combining offset vectors and an attention mechanism, offset-attention can better capture and process local geometric information in point clouds, improving the performance and accuracy of corn point cloud completion tasks. An offset-attention module calculates the offset between the input features and the attention features via element-wise subtraction. Offset-attention leverages the robustness of relative coordinates in transformations and the effectiveness of the Laplacian matrix in convolution, specifically as shown in the following equation:

$$F_{out} = OA(F_{in}) = LBR(F_{in} - F_{sa}) + F_{in} \quad (6)$$

where $F_{in} - F_{sa}$ is analogous to a discrete Laplacian operator; then, it is passed to the LBR (combining the Linear, BatchNorm, and ReLU layers) network layer and combined with the input features to obtain the output features of offset-attention.

Xie proposed the original GRNet for completing regular man-made objects, such as airplanes, tables, chairs, and lamps (2020). Compared with these regular man-made objects, corn leaves are thin and irregular, with curled leaves, which results in difficulties in completion. So, we explored whether MGRNet+OA could be used to complete the LiDAR points of corn plants in this study.

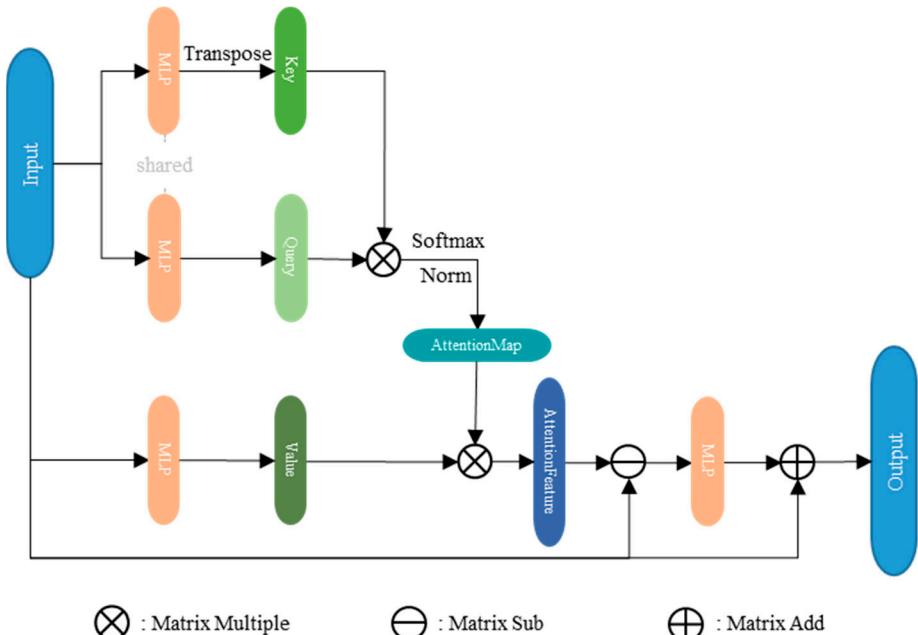


Figure 3. The detailed structure of the offset-attention module. \otimes : matrix multiplication; \ominus : matrix subtraction; and \oplus : matrix sum.

2.2. Construction of Loss Function

For unorganized characteristics, neither the cross-entropy method of 3D grids nor the L1/L2 loss of a 2D image can be used to complete LiDAR points directly. Therefore, the chamfer distance has been used as the loss function to train the neural networks of 3D studying models [48]. The chamfer distance (lower is better) is calculated as follows:

$$CD = \frac{1}{n_{\mathcal{T}}} \sum_{t \in \mathcal{T}} \min_{r \in \mathcal{R}} \|t - r\|_2^2 + \frac{1}{n_{\mathcal{R}}} \sum_{r \in \mathcal{R}} \min_{t \in \mathcal{T}} \|t - r\|_2^2 \quad (7)$$

where $\mathcal{T} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathcal{T}}}$ and $\mathcal{R} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathcal{R}}}$ denote the output result and the ground truth, respectively.

The reason why the chamfer distance cannot ensure that the predicted points are consistent with the geometric features of objects is that the combination of the chamfer distance and the gridding loss is used to validate the completion of LiDAR points from corn plants. The gridding loss is the L1 distance between two 3D grid datasets (Figure 4), and its expression is determined as follows:

$$\mathcal{L}_{Gridding}(W^{pred}, W^{gt}) = \frac{1}{N_G^3} \sum \|W^{pred} - W^{gt}\| \quad (8)$$

where $G_{pred} = \langle V^{pred}, W^{pred} \rangle$ is the predicted 3D grid, $G_{gt} = \langle V^{gt}, W^{gt} \rangle$ is the real 3D grid generated from LiDAR points, and $W^{pred} \in \mathbb{R}^{N_G^3}$, $W^{gt} \in \mathbb{R}^{N_G^3}$; N_G is the resolution of the two 3D grids.

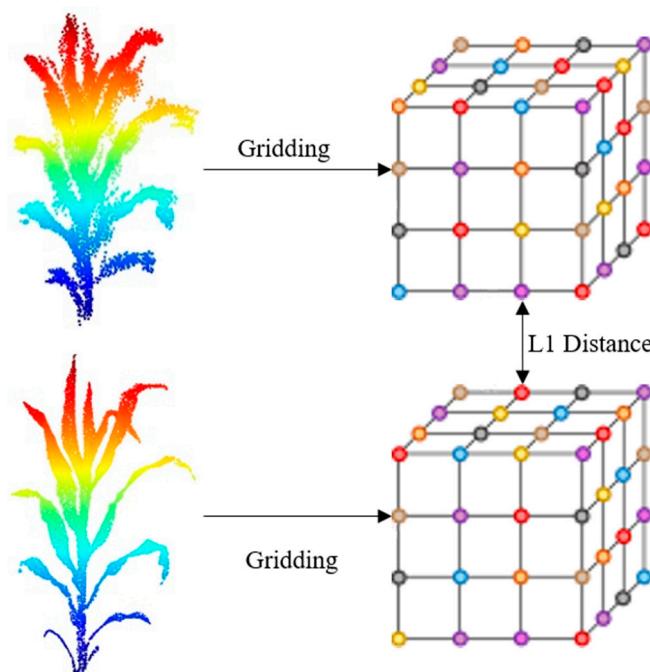


Figure 4. Diagram of gridding loss by L1 distance. This image shows the L1 distance between the gridding of the predicted points and the gridding of the ground truth.

3. Experiments

3.1. Collection and Pre-Processing of Terrestrial LiDAR Point Cloud

3.1.1. Collection of Terrestrial LiDAR Points

The terrestrial LiDAR data used for completion in this study were collected using a Trimble TX8 3D laser scanner on a farm in Huailai City, Hebei Province, China. The data collection dates were 16 September, 25 September, and 8 October 2018, and the growing stages were the seventh-leaf stage, the jointing stage, and the flare-opening stage, respectively. Trimble TX8 is an impulse TLS that collects LiDAR points at a wavelength of $1.5 \mu\text{m}$ with a scanning rate of 1MHz and a field of view of $360^\circ \times 317^\circ$. The maximum scanning range is 120 m with a system error of less than 2 mm. The angular resolution is $\pm 8''$ for scanning, and the minimum point spacing is 15.1 mm at a 10 m distance. The scanner was mounted on a tripod and erected at a height of about 1.5m in this study. To collect the complete LiDAR points of dense corn plants, we collected data from three to four stations for every experiment. And there, four white target balls were used to conduct geometric registration between different stations. Each station was scanned for about 10–30 min.

3.1.2. Pre-Processing of Terrestrial LiDAR Points

The corn point cloud was collected during three growing seasons in this study. The geometric registration of the point cloud from different stations was performed using Trimble Realworks Survey software. The point cloud data obtained from a single station were significantly missing; more complete field corn point cloud data were obtained by registering the results of four stations. And the geometric registration results for corn plants in the seven-leaf season are shown in Figure 5. The point cloud data obtained from a single station was affected by occlusion among the target objects, which led to serious deficiencies in the results. After the geometric registration of the scanning results from different angles, more complete point cloud data could be acquired.

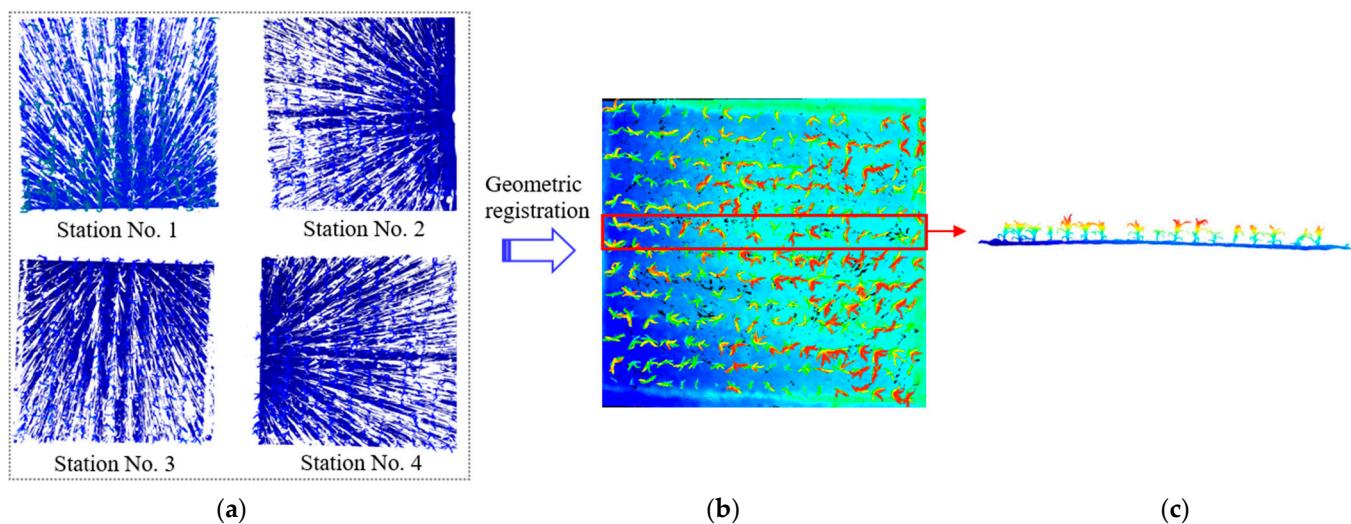


Figure 5. An example of geometric correction for the LiDAR points of corn plants: (a) scanning at four different orientation stations; (b) top view after geometric registration; and (c) the front view of one row after geometric registration.

To improve the data quality, the point cloud was preprocessed, specifically including the separation of individual corn plant points, the filtering of ground points, and the removal of noise. Furthermore, the down-sampling of dense points was performed to avoid increasing training work, and normalization was applied to improve the convergence of the 3D CNN network. And the normalization was conducted as follows:

$$(x, y, z) = \frac{(x, y, z) - \text{mean}(x, y, z)}{\max(x, y, z) - \min(x, y, z)} \quad (9)$$

where (x, y, z) is the 3D coordinate of corn leaves' LiDAR points and n is the number of points. All the corn plant samples needed to be normalized, so the central coordinates of each sample were normalized to the origin of the coordinate with the same range of $[-1, 1]$.

The corn plants' shapes varied with the different varieties in different growing seasons. This difference would affect the training efficiency and accuracy of the large studding distance resulting from the large difference and disorder pattern of corn plants. Therefore, preregistration should be performed for training after normalization. The corn stalks were adjusted to the same height and position (Figure 6e). The length, angle, and number of the corn leaves were varied, too. So, the main plane of the corn plants was adjusted into the same plane with a similar leaf azimuth (Figure 6d–f).

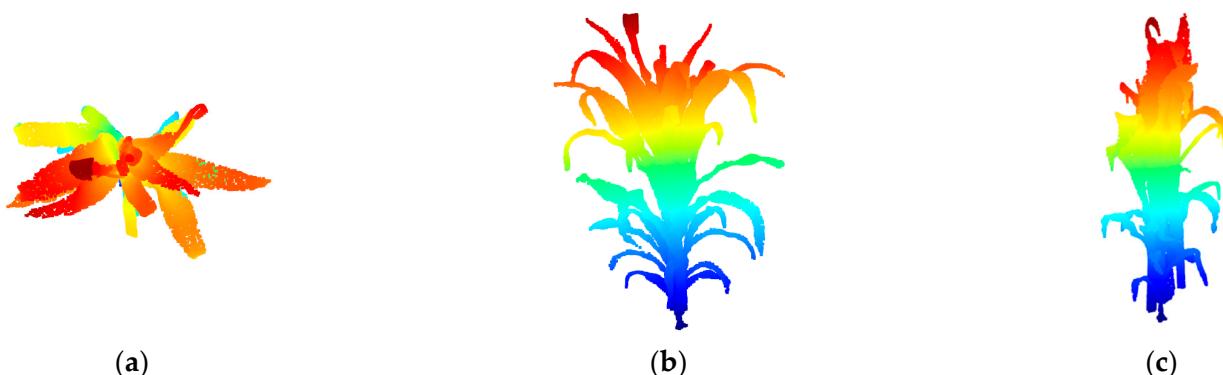


Figure 6. Cont.

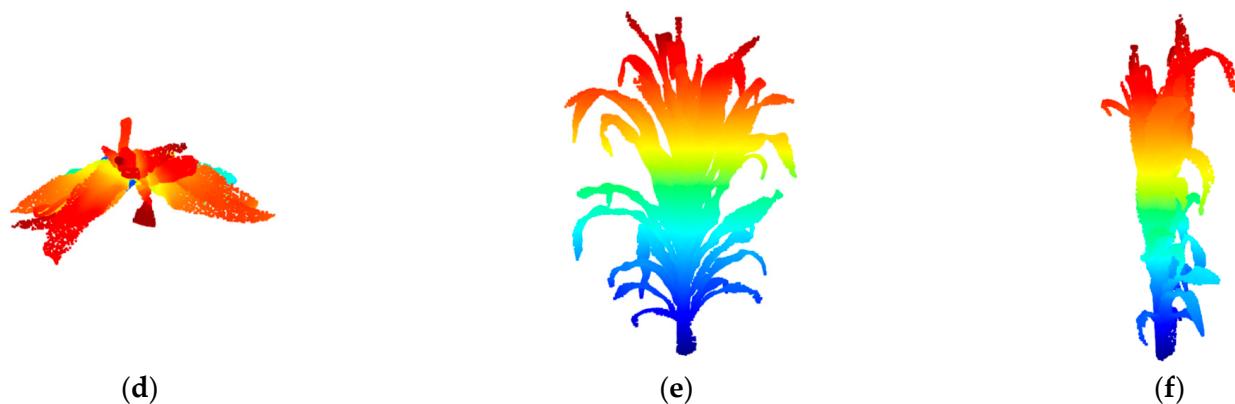


Figure 6. The pre-alignment of corn plants' points: (a–c) top view, front view, and side view of pre-processes points; (d–f) the top view, front view, and side view of pre-aligned points.

3.2. Generation of a Training and Testing Dataset

Compared with the man-made regular airplane and chair samples shared in the ShapeNet database, the completion of corn plants' points is more difficult due to their thin, irregular leaves. Therefore, we built a corn leaf LiDAR point dataset using plants with different varieties in three different growing seasons—the seven-leaf stage, jointing stage, and trumpet stage—which were used to train and validate the 3D CNN networks. There were 137 corn plants used to generate the sample's dataset, with 77 plants at the seven-leaf stage, 40 plants at the jointing stage, and 20 plants at the trumpet stage. During the data collection process, the dense shading of the corn leaves at the trumpet stage caused poor collection, and fewer usable samples were screened. Every sampling plant was augmented to 15–20 sampling plants with incomplete LiDAR points using CloudCompare software (<https://www.cloudcompare.org/>) (accessed on 20 April 2021)), as shown in Figure 7. To improve training efficiency, down-sampling was performed for every sampling plant with a total number of points of about 5000. Figure 8 presents an example of the training and testing dataset of incomplete ((f) to (j)) and complete ((a) to (e)) corn plants. All the sampling plants were divided into a training dataset and a testing dataset with a total number of 962 (from 137 corn plants) through data augmentation to reduce the manual workload and maintain balanced data distribution of the corn samples from the three growth stages. For the ground truth samples, a small positional offset (1–5% of stem height) was added along the corn stem, and the process was repeated in 5 to 15 iterations for every corn plant. The point number of a complete corn plant in the dataset was 16,384, and that of an incomplete corn plant was around 5000.

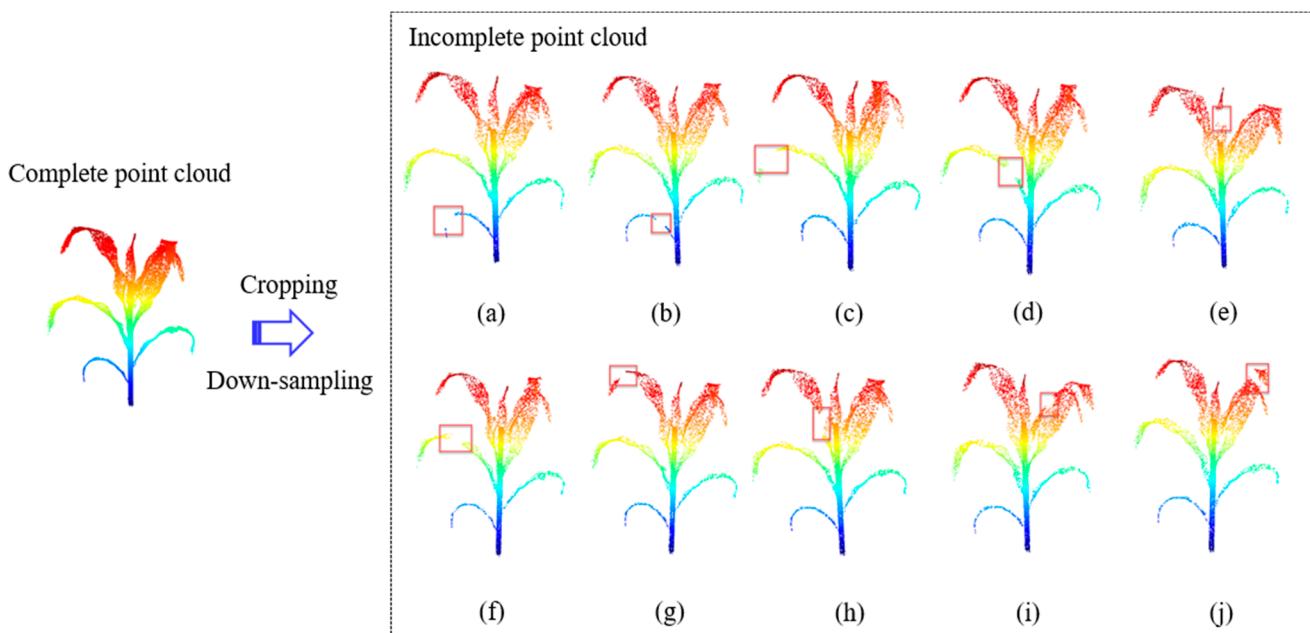


Figure 7. An example of the training and testing dataset for the incomplete (**a–j**) corn plants.

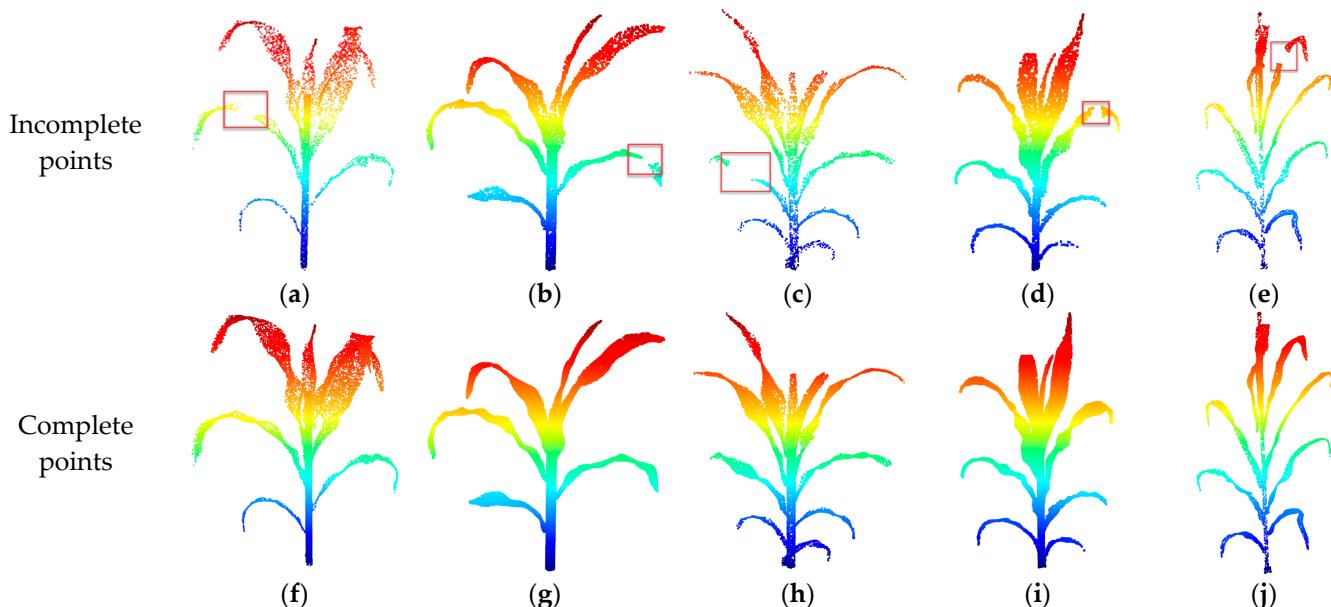


Figure 8. The visualization results of the complete point cloud and the corresponding incomplete point cloud: (**a–e**) the visualization results of the incomplete point clouds and (**f–j**) the visualization results of the complete point clouds.

3.3. Training and Validation of MSGRNet+OA

The completion of corn plants' points was achieved by training the 3D structural characteristics of the complete and incomplete points of corn plants. And the experiments were conducted on a desktop computer with an Intel(R) Xeon(R) Gold 6330 processor (2.00 GHz, 42 MB cache) CPU, 160 GB of RAM, and an NVIDIA RTX3090 (24 GB) GPU. The network was trained using the PyTorch1.9.0 framework and CUDA11.1 with GPU acceleration. All the training samples were 3D LiDAR point data with 3D (x,y,z) coordinates, and the number of points in a sample's complete point cloud was fixed at 16,384 points. The Adam optimizer [49] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ was used for training. The initial learning rate was set to 10^{-4} , which was decreased by half after 50 iterations. The training

procedure continued for a total of 300 iterations with a decreasing loss function. And the iteration started to converge after 100 iterations.

Since there was not enough complete corn point cloud data, we applied transfer learning based on a pre-trained model to improve the training effectiveness of the model. A subset of the corn point cloud data was selected for pre-training. Then, the pre-trained model was used as the initial model for the task, and the parameters of the pre-trained model were used as the initial parameters for the target task.

The testing process used the same network acquired from the training process, and the batch size was fixed at 1. The number of each training corn sample was down-sampled to 16,384 points during validation. During testing, the completed points were compared with the complete ground truth points. The chamfer distance and F-score@1% were used to validate the completion accuracy. The calculation of the chamfer distance is described in Section 2.2. And the F-score@1% was defined according to Tatarchenko et al. [31], as follows:

$$\text{F-Score}(d) = \frac{2P(d)R(d)}{P(d) + R(d)} \quad (10)$$

where $P(d)$ is the accuracy for a given distance threshold, and $R(d)$ is the recall for a given distance threshold. The calculations of $P(d)$ and $R(d)$ were as follows:

$$P(d) = \frac{1}{n_{\mathcal{R}}} \sum_{r \in \mathcal{R}} \left[\min_{t \in \mathcal{T}} \|t - r\| < d \right] \quad (11)$$

$$R(d) = \frac{1}{n_{\mathcal{T}}} \sum_{t \in \mathcal{T}} \left[\min_{r \in \mathcal{R}} \|t - r\| < d \right] \quad (12)$$

where $\mathcal{T} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathcal{T}}}$ is the ground truth, $\mathcal{R} = \{(x_i, y_i, z_i)\}_{i=1}^{n_{\mathcal{R}}}$ is the reconstructed point set being evaluated, $n_{\mathcal{T}}$ and $n_{\mathcal{R}}$ are the numbers of points of \mathcal{T} and \mathcal{R} , respectively, and d is a distance threshold.

4. Completed Results and Analysis

4.1. Completed Results of Different Improvement Methods

Given the difficulties of corn completion, there were two improvements to multi-scale 3D CNN and offset-attention that were integrated to improve the performance of GRNet. To evaluate the impact of the proposed strategy on model performance quantitatively and qualitatively, there were four ablation experiments that were designed, which are illustrated in Table 1. In Table 1, none of the improvement strategies was used in the GRNet model; GRNet+OA applied the improvement strategies for offset-attention. Multi-scale 3D CNN was applied to improve MSGRNet. The improvement strategies for multi-scale 3D CNN and offset-attention were applied to MSGRNet+OA. To evaluate the impact of the number of offset-attention additions on the MSGRNet+OA network completion results, ablation experiments were performed, varying the number of offset-attention modules. When the number of OAs was 1 or 2, this CD ($\times 10^{-4}$) was 1.302 and 1.258, and the F-score@1% was 0.840 and 0.843, respectively. However, when the number of OAs was greater than 2, the GPU memory of our computer could not support the training task of the network, and the GPU memory required was beyond 24 G.

Table 1. Comparison of ablation experiments with different improvement methods.

Method	Metrics
	CD ($\times 10^{-4}$)
GRNet	1.439
GRNet+OA	1.360
MSGRNet	1.327
MSGRNet+OA	1.258
	F-Score@1%
GRNet	0.825
GRNet+OA	0.836
MSGRNet	0.838
MSGRNet+OA	0.843

According to the results in Table 1, the MGRNet+OA method proposed in the experiments performed the best, with 1.258 and 0.843, respectively, for the average CD ($\times 10^{-4}$) value and the F-score@1%. A comparison of the results of the ablation experiments with a modified number of OAs demonstrated that MGRNet+OA had the best complementary effect when the number of OAs was set to 2 in our computational context.

4.2. Completed Results of Different Training Scenarios

4.2.1. Completed Results Using Different Batch Sizes

To determine whether the batch size had an effect on the complete results, we compared the complete results with batch sizes of 4, 8, 16, 32, and 64. And all these experiments were performed for 300 epochs. Figure 9 reveals that the completion accuracy for a batch size of 32 was higher than that of the other batch sizes with a CD ($\times 10^{-4}$) of 1.258 and an F-score@1% of 0.843, respectively. Furthermore, the training time for a batch size of 32 was the shortest at 11.417 s in all the comparison experiments. As is well known, the training duration is long, and the fluctuation of the gradient within an adjacent iteration is large when the batch size is small. On the contrary, too large a batch size leads to a slight change in the gradient and, thus, an unreasonable local minimum. Therefore, the batch size of 32 is optimal for corn plant completion, given the training efficiency, duration, and GPU memory in this study.

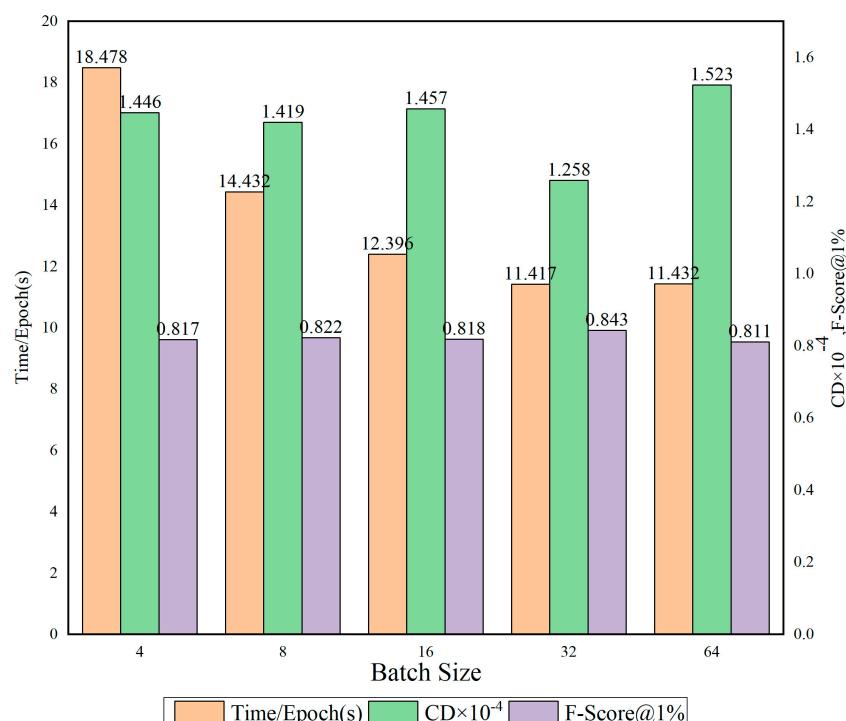


Figure 9. Completion accuracy using different batch sizes. The accuracy assessment indices included time/epoch(s), CD $\times 10^{-4}$ (lower is better), and F-Score@1% (higher is better) using different batch sizes.

4.2.2. Completed Results Using Different Training/Testing Ratios

Typically, a training set is used for parameter updating and the optimization of a model, while the test set is used to evaluate the model's ability to generalize from unseen data. Therefore, the ratios of training and test sets can affect the accuracy and performance of a model during training and testing. In this study, five ablation experiments were set up to determine the most suitable training/testing ratio for the network. The completed results using different ratios of training and testing samples were compared to optimize the completion experiment. This comparison was performed using training/testing ratios

of 60:40, 70:30, 75:25, 80:20, and 90:10, respectively. There were 962 corn plant LiDAR point samples used for the comparison experiment in total. Figure 10 reveals that the accuracy of completion using a training/testing ratio of 90:10 was the highest with a $CD (\times 10^{-4})$ of 1.258 and an F-score@1% of 0.843.

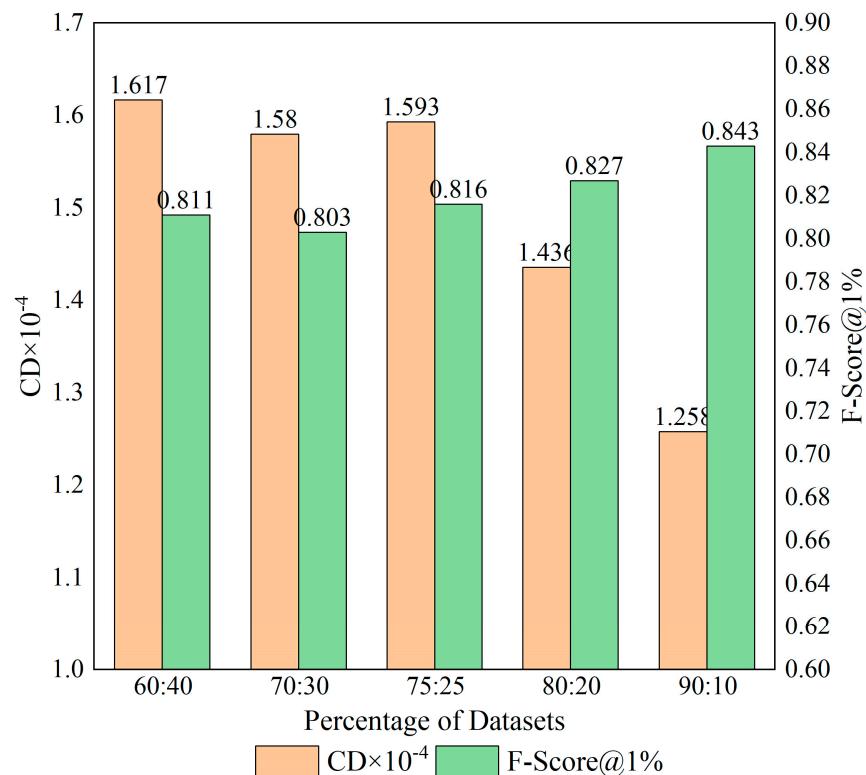


Figure 10. Completion accuracy using different training/testing ratios. The accuracy assessment indices include a $CD (\times 10^{-4})$ (lower is better) and an F-score@1% (higher is better) using different training/testing ratios.

4.2.3. Completed Results Using Different Grid Scales

The size of the grid scale will have some impact on the network training results. A larger grid scale will increase the computational complexity; a smaller grid scale will lead to a loss of information and, thus, reduce the feature extraction ability of the network, while a proper grid scale can better capture the structure and features of the input data. Therefore, the optimal grid size was chosen based on experiments and tuning to improve the performance and generalization ability of the model. We compared the completed results using grid scales of 32, 64, and 128, and other network parameters were set consistently. Figure 11 shows that the accuracy of completion using a grid scale of 64 was the highest with a $CD (\times 10^{-4})$ of 1.258 and an F-score@1% of 0.843. With a grid scale of 32, the scale was too small to lose data information significantly; with a grid scale of 128, the scale was too large, the layers of abstraction of the data information in the network were not enough, and the amount of the calculation was greater. Therefore, a grid scale of 64 is prime for corn plants' completion.

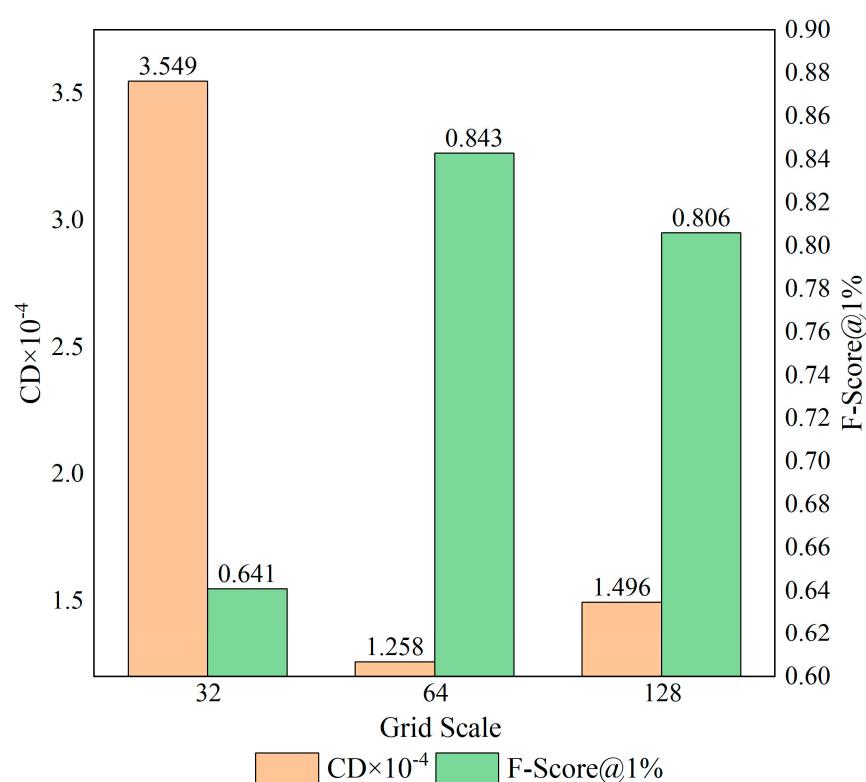


Figure 11. Completion accuracy using different grid scales. The accuracy assessment indices include a CD ($\times 10^{-4}$) (lower is better) and an F-score@1% (higher is better) using different grid scales.

4.3. Completed Results in Different Growing Seasons

The leaf number, leaf angle, and 3D structural traits of corn plants in different growing seasons are quite different. So, we compared the completed results of corn plant points in the seven-leaf stage, the jointing stage, and the trumpet stage. The visualization results are shown in Figure 12. They reveal that the complete result in the seven-leaf stage was better than that in the other two growing seasons. Figure 12b presents the completed results of the seven-leaf stage. Comparably, the number of corn leaves was low, and the 3D structure was relatively simple in the seven-leaf stage. And the complete result was acceptable. Figure 12e shows the completed result in the jointing stage. In this stage, the number of corn leaves increased, and the 3D structure of corn was more complicated compared with that at the seven-leaf stage. So, the completed results of corn leaves' points in the jointing stage were not as good as those in the seven-leaf stage. Figure 12h shows the completed result of the trumpet stage. The number of corn leaves increased further, and the 3D structure of corn was more complicated with the decreased gap and increased overlapping between corn plants. The completed results were worse than the results in the two growing seasons, especially for the small lower leaves. In addition, with the growth of corn, the structure of corn becomes more complex, increasing the difficulty of data collection. As a result, the complete data on corn in the dataset for the period of jointing and the trumpet period were relatively few, which also led to the poor results of the complementation of corn during these two periods.

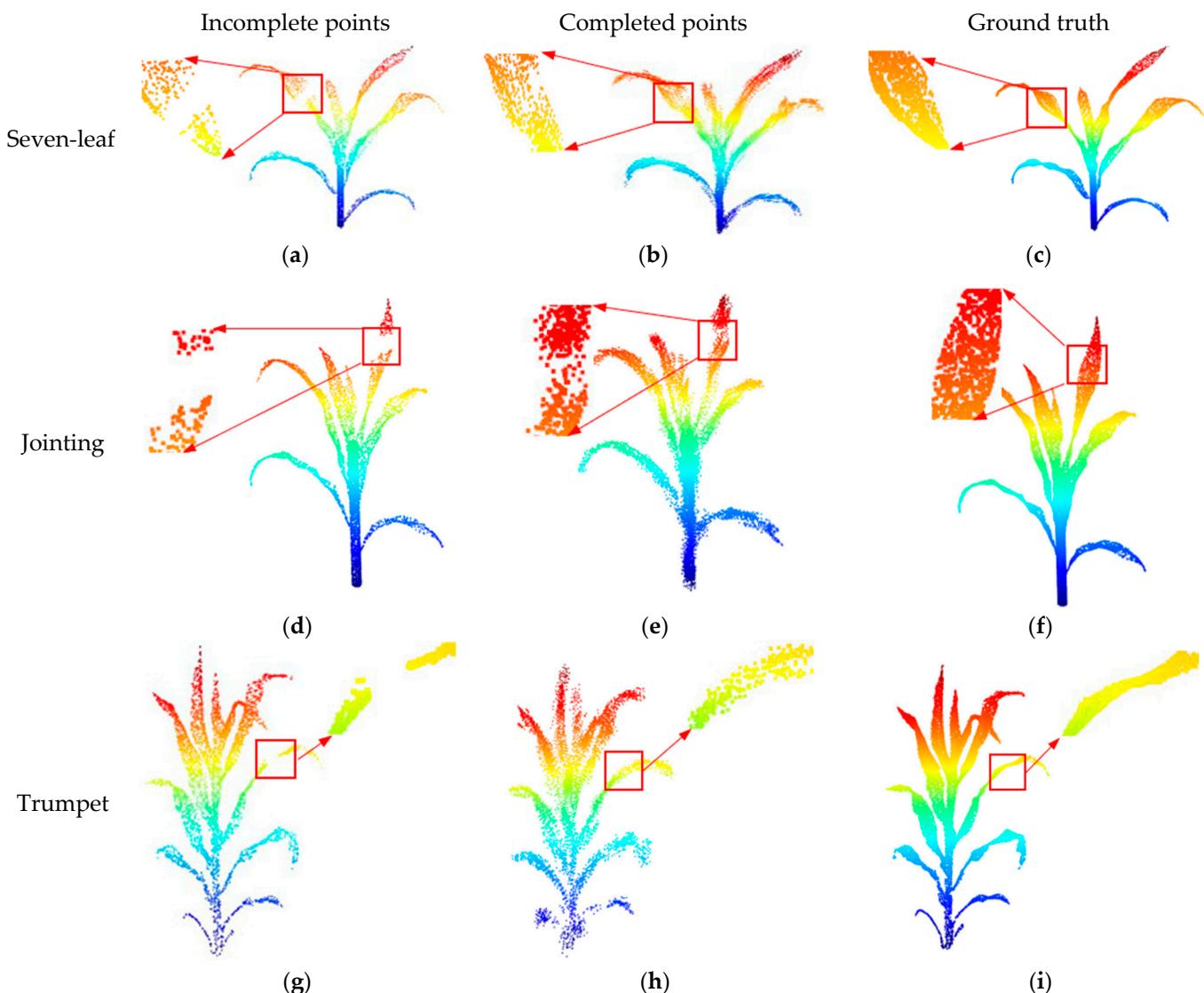


Figure 12. Comparison of completed results in three growing seasons: (a–c) the incomplete point clouds, complete point clouds, and ground truth at the seven-leaf stage, respectively; (d–f) the incomplete point clouds, complete point clouds, and ground truth at the jointing stage, respectively; and (g–i) the incomplete point clouds, complete point clouds, and ground truth at the trumpet stage, respectively.

The quantitative testing results also revealed this finding with the CD ($\times 10^{-4}$) and F-score@1% in the seven-leaf stage, the jointing stage, and the trumpet stage at 2.621 and 0.684, 2.649 and 0.667, and 2.743 and 0.427, respectively. Generally speaking, the completion of corn leaves in the trumpet stage was better than the result in the seven-leaf stage for large leaves. Unfortunately, limited by the total of 16,384 points for each completion, the points' density and the fine details of corn leaves in the seven-leaf stage were better than those in the trumpet stage. Therefore, the completion accuracy in the trumpet stage was not higher than that in the seven-leaf stage. This limitation of MGRNet+OA for completion should be solved in future research.

4.4. Comparison with Other Completion Methods

To evaluate the efficiency of the network for completing corn plants, we compared the completed results with the results using PCN, shape inversion, the original GRNet, SeedFormer, and the PMP-Net++ approach. The PCN network model is an encoder/decoder-

based network that operates directly on the original point cloud, which is a baseline model for point cloud completion [32]. Its encoder encodes the structural information as the eigenvectors for the training of input LiDAR points' data. Then, the 3D global features and local features of corn plants are acquired via the multi-layer perceptron and the maximum pooling layer. Lastly, the decoder transforms the output eigenvectors of the encoder into the output points. Shape inversion is an inverse mapping method of the generative adversarial network (GAN) [50], which transforms the input data into potential space in a pre-trained GAN model. Then, the reverse coding of the generator is used for completion. This network updates the potential space by back-propagating the loss function through a descent gradient method. Then, the trained generated adversarial network is fine-tuned to output the accurate points. SeedFormer is a method based on the Transformer architecture, combining self-attention and fully connected layers to capture information from local and global contexts in point clouds. It generates initial seed points by constructing local subgraphs and utilizes the self-attention mechanism of transformers to adaptively update and refine these seed points, resulting in the final seed points. This mechanism enables the model to perform global optimization and inference while considering local information simultaneously. PMP-Net++ is an advanced and extended version of PMP-Net that introduces a mutually reinforcing mechanism for the interaction between global and local information. It provides an effective way to improve the performance of the feature representation of point clouds.

The quantitative metrics of the $CD (\times 10^{-4})$, $F\text{-score}@1\%$, $Params$, and $GFLOPS$ are used to evaluate the completed accuracy of PCN, shape inversion, the original GRNet, and MGRNet+OA. The $CD (\times 10^{-4})$, $F\text{-score}@1\%$, $Params$, and $GFLOPS$ of completed corn plant points using PCN, shape inversion, GRNet, and MGRNet+OA are shown in Table 2. Our results reveal that, compared with PCN, shape inversion, the original GRNet, SeedFormer, and PMP-Net++, the $CD (\times 10^{-4})$ was reduced by 15.882, 15.960, 0.181, 1.852, and 1.471, and the $F\text{-score}$ was improved by 0.404, 0.450, 0.018, 0.274, and 0.203, respectively. Therefore, we could conclude that our network performs the best completion of corn plant points compared with the completed results of the PCN, shape inversion, original GRNet, SeedFormer, and PMP-Net++ methods. In addition to accuracy, efficiency is also an important metric for evaluating deep learning networks. We compared the network parameters and GFLOPS of these methods. The comparison results show that the original GRNet has the largest number of network parameters and GFLOPS, and the improved MGRNet+OA has a reduced number of network parameters and GFLOPS, but these are still larger than those of the other methods.

Table 2. Comparison with other completion methods evaluated by $CD \times 10^{-4}$ (lower is better), $F\text{-score}@1\%$ (higher is better), $Params$, and $GFLOPS$.

Method	Metrics			
	$CD (\times 10^{-4})$	$F\text{-Score}@1\%$	$Params$	$GFLOPS$
PCN	17.140	0.439	6.864	14.708
Shape Inversion	17.218	0.393	5.653	10.465
GRNet	1.439	0.825	76.708	25.881
SeedFormer	3.110	0.569	3.3118	6.917
PMP-Net++	2.729	0.640	5.888	4.503
MGRNet+OA	1.258	0.843	61.174	23.671

In addition, a visualization evaluation was performed to compare the completed results of corn plant points using PCN, shape inversion, the original GRNet, SeedFormer, PMP-Net++, and MGRNet+OA, as shown in Figure 13. The visualization of the complementary results revealed that the completed result using MGRNet+OA was better than that of PCN, shape inversion, GRNet, SeedFormer, and PMP-Net++ with the full corn leaves and stalk and fine details of whole corn plants. There was obvious ambiguity and varied misplacement for the completed corn leaves using PCN, and we almost could not

distinguish between corn plants. The points were mainly concentrated in the middle stalk of the corn plants, and there was almost no recognized leaf shape. For the completed result using the shape inversion method, there were clearer corn leaf shapes and recognized plants. However, the completed results for the boundary of corn leaves were still blurred. As seen in the results of SeedFormer’s visual complementation, it was able to predict the corn structure, but it was not sufficient to complement the missing parts of the corn. The complementation results of PMP-Net++ demonstrated its ability in the corn point cloud complementation task, but the complementation effect was less than ideal, and there were also point movement anomalies. Moreover, the problem of the loss of edge information and the reconstruction of tiny parts could be alleviated using MGRNet+OA compared to the original GRNet.

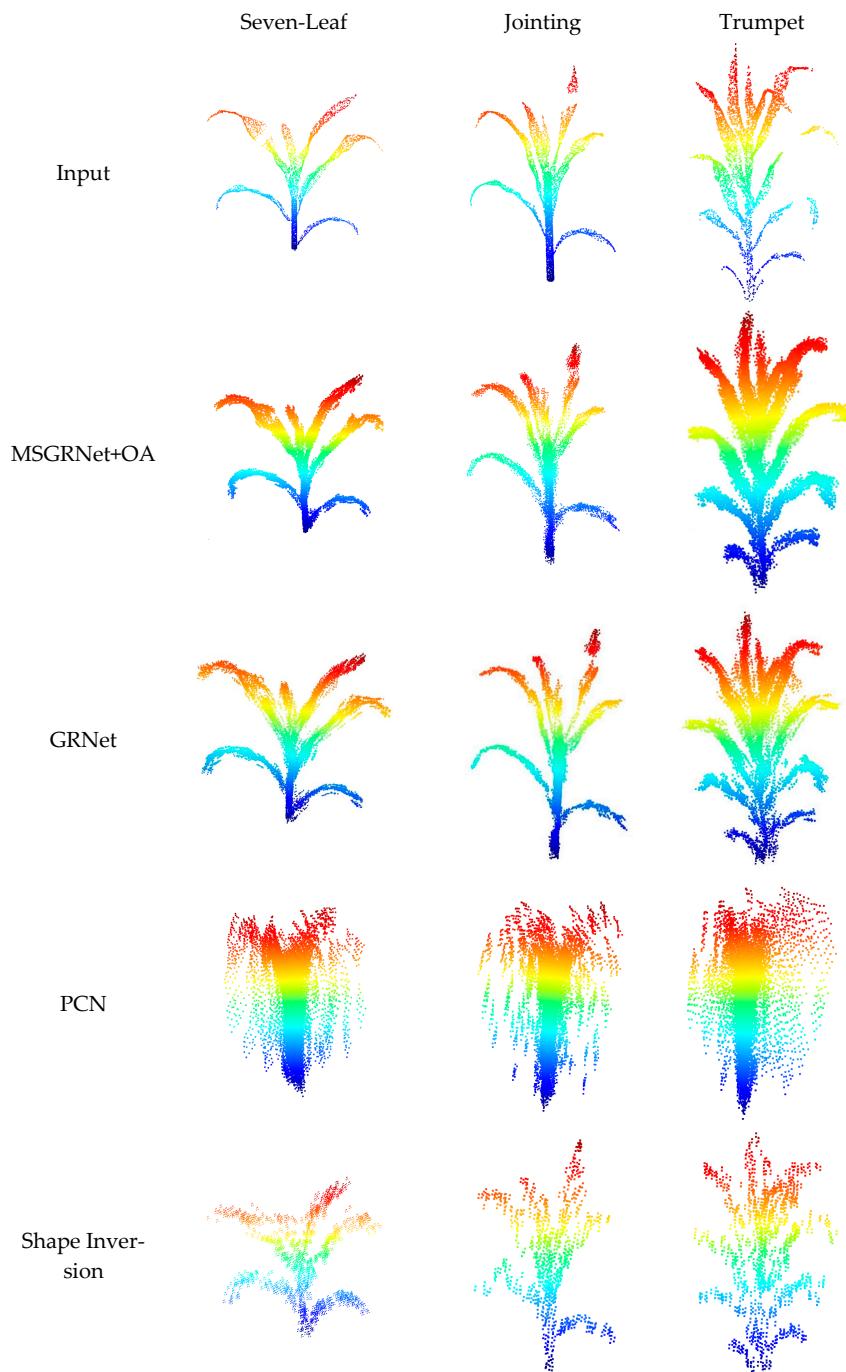


Figure 13. Cont.

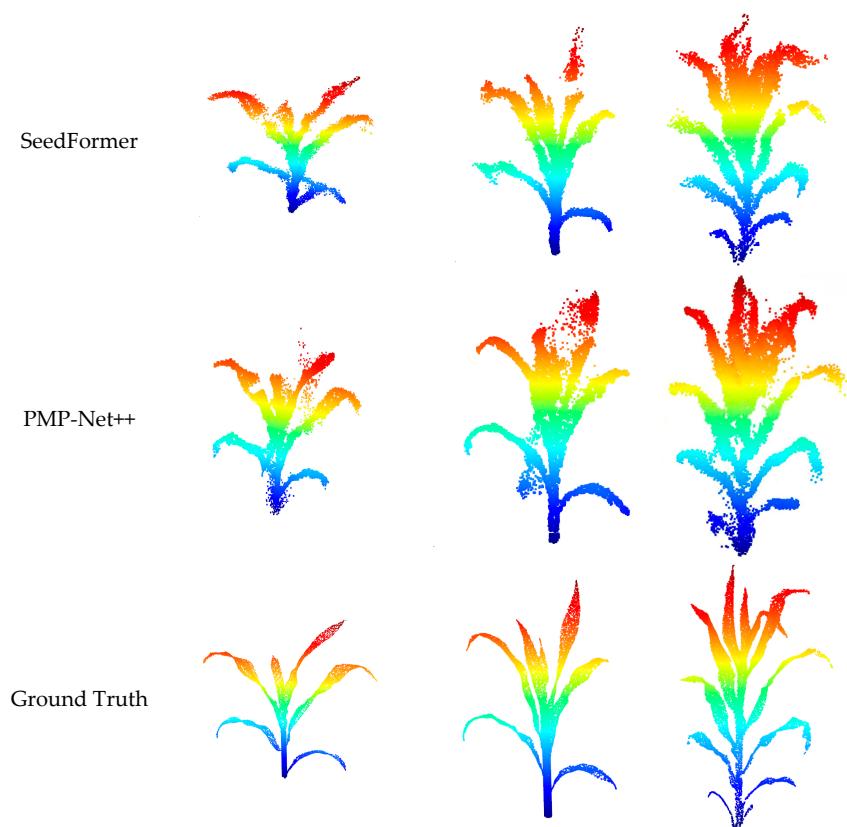


Figure 13. Comparison of the completed results of corn plants in three growing seasons using the MSGRNet+OA, GRNet, PCN, and shape inversion methods.

In summary, MSGRNet+OA performed quantitatively and qualitatively better than PCN, shape inversion, the original GRNet, SeedFormer, and PMP-Net++ in completing corn plant points. Since PCN was developed based on original points, uniformly distributed points were generated to reduce the chamfer distance. Therefore, many details were missing in the completed results of the PCN method. The shape inversion method does not require a complete point cloud during training to complete the point cloud because it is an unsupervised learning network. So, it is easy to create a sample dataset for training. Unfortunately, the incomplete samples led to a lack of loss calculation to correct the training results. And the completed results of the corn plants were also fuzzy and fluctuating. The SeedFormer model uses a seed point generation mechanism to select representative points to guide the completion task. However, due to the irregular structure of corn leaves, the generated seed points did not provide satisfactory results and could not effectively perform the task of completing the corn point cloud. The PMP-Net++ introduces the point-wise mutual promotion mechanism to improve the feature representation and semantic segmentation performance at the point level, the local level, and the global level. Through a two-stage training strategy, it progressively optimizes global and local features to improve the accuracy of corn point cloud completion tasks. PMP-Net++ is a non-generative network that completes missing parts by shifting the original point cloud, and it performs poorly when the point cloud is highly incomplete. PMP-Net++ completes the missing parts by shifting the original points and providing the non-missing parts of the original data, so it performs better in terms of the accuracy of its complementary results, but it performs worse in terms of visual dialog results. Meanwhile, MSGRNet+OA introduces a multi-scale mechanism and offset-attention, and it converts the point cloud into a grid structure with structural information, which led to better results for complex and irregular corn leaves. Comparatively, the result of corn plant point completion using MSGRNet+OA was the most effective of these six methods with a more detailed corn plant structure even though the corn leaves were thin and curled. Our modified network has the potential for plant

point completion. On the other hand, considering the efficiency of the network, the results of the comparison between the number of network parameters and GFLOPS show that the efficiency of the improved MSGRNet+OA is less than that of the original GRNet but still greater than that of the other networks. This indicates that the computational cost of MSGRNet is higher than that of the other networks and that the computational efficiency is lower than that of the other networks. This indicates that MSGRNet+OA still needs improvement in terms of network efficiency.

5. Discussion

The completion of thin, curled corn leaves was performed in this study, which is more difficult than the completion of man-made objects with regular shapes, such as airplanes, chairs, lamps, and tables. Therefore, we developed MSGRNet+OA to complete corn plant points by transforming the point cloud into a 3D grid and normalizing the unorganized points using the 3D grid as an intermediate representation. We introduced multi-scale prediction fusion and incorporated an offset-attention mechanism. This operation was performed to acquire more detailed structural and contextual information on corn plants. Comparisons of the completed results with the different training parameters of the network, different growing seasons, and different deep learning methods were performed. Our results reveal that MSGRNet+OA has the potential to complete the points of vegetation. This study serves as a reference for the completion of other crop plants, which is meaningful for the phenotyping of crops with high precision.

Certainly, there were also some limitations in this study. Firstly, only corn plants were used to conduct the completing experiments, excluding other crops. And the completion of other grain crops, vegetables, and fruit trees should be done in the future. Secondly, a large number of samples is required for the training of 3D deep learning models. More samples of different plants with different structures and varieties must be created and shared for 3D deep learning studies. Lastly, further optimization of the network and its loss function are needed to improve the completion of crop plants and address the disadvantages resulting from small, irregular, and curled leaves. Therefore, future work should include improving the generalization capability of the MSGRNet+OA model, expanding the crop types and varieties, and augmenting the samples for deep learning.

6. Conclusions

This study has focused on exploring the potential of MSGRNet+OA for completing the 3D point cloud of thin corn plants. A dataset of 962 sampling LiDAR points of corn plants with different varieties in three growing seasons was built firstly, which is shared along with this study. The comparison of the completed results using different improvement methods' different batch sizes, training/testing ratios, and grid scales and the completed results in different growing seasons was performed to optimize the completed experiments. Comparisons of MSGRNet+OA with PCN, shape inversion, the original GRNet, SeedFormer, and PMP-Net++ were performed to validate the efficiency of Multi-scale 3D CNN deep learning methods. The conclusions are as follows:

- (1) For the parameter setting of corn plants' completion, a batch size of 32 and a grid scale of 64 are optimal for accurate completion using a training/testing ratio of 90:10 with the highest CD ($\times 10^{-4}$) of 1.360 and F-score@1% of 0.836.
- (2) The completion accuracy in different growing seasons is varied. Our results reveal that the complete result in the seven-leaf stage is better than that in the other two growing seasons.
- (3) MSGRNet+OA is the most effective approach when compared with the PCN, shape inversion, GRNet, SeedFormer, and PMP-Net++ methods with a low CD ($\times 10^{-4}$) of 1.258 and high F-score@1% of 0.843.

In the future, we will extend the dataset of plant samples to build more robust 3D CNN deep learning networks. Simultaneously, in response to the complex and irregular characteristics of plant structures, we will research more efficient network mechanisms to

enhance the richness of feature extraction, preserve details and structures, and improve network robustness. We believe this study will contribute to the development of the accurate, high-throughput phenotyping of crops. Moreover, many studies are currently exploring the potential of graph neural networks in point cloud applications, and some studies have revealed that using graph neural networks to represent point clouds offers more advantages than neural networks based on multi-layer perceptrons. Graph neural networks can capture the relationships and global structures between points in point cloud data, enabling them to learn local and global features for point cloud completion. By constructing a graph structure and using graph convolutional operations, graph neural networks can understand the local and global contexts of a point cloud, allowing them to generate missing points and complete the point cloud. However, there is a possibility of losing substructure information during the process of converting point clouds into graphs. Therefore, a future research direction for us is to preserve the structural information of point cloud data while utilizing graphs as an intermediate representation for point cloud completion tasks.

Author Contributions: This work involved cooperation among our research team, and the contributions are as follows: conceptualization, Y.Z. and W.S.; methodology, Y.Z.; software, W.T. and X.H.; validation, W.S. and W.T.; data curation, Y.Z. and Z.L.; writing—original draft preparation, Y.Z.; writing—review and editing, W.S.; visualization, Z.Z. and C.X.; supervision, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number No. 42171331, and the 2115 Talent Development Program of China Agricultural University.

Data Availability Statement: <https://doi.org/10.6084/m9.figshare.21384186> (accessed on 23 October 2022).

Acknowledgments: We appreciate the editors and anonymous reviewers for their valuable time, constructive suggestions, and insightful comments.

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

References

1. Lefsky, M.A.; Hudak, A.T.; Cohen, W.B.; Acker, S.A. Geographic variability in lidar predictions of forest stand structure in the Pacific Northwest. *Remote Sens. Environ.* **2005**, *95*, 532–548. [[CrossRef](#)]
2. Hoffmeister, D.; Curdt, C.; Tilly, N.; Bendig, J. 3D Terrestrial Laser Scanning for Field Crop Modelling. In Proceedings of the ISPRS WG VII/5 Workshop on Remote Sensing Methods for Change Detection and Process Modelling, Cologne, Germany, 18–19 November 2010.
3. Guo, Q.; Wu, F.; Pang, S.; Zhao, X.; Chen, L.; Liu, J.; Xue, B.; Xu, G.; Li, L.; Jing, H. Crop 3D—A LiDAR based platform for 3D high-throughput crop phenotyping. *Sci. China Life Sci.* **2018**, *61*, 328–339. [[CrossRef](#)] [[PubMed](#)]
4. Furbank, R.T.; Tester, M. Phenomics—Technologies to relieve the phenotyping bottleneck. *Trends Plant Sci.* **2011**, *16*, 635–644. [[CrossRef](#)] [[PubMed](#)]
5. Liang, X.; Kankare, V.; Hyypä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; Yu, X.; Kaartinen, H.; Jaakkola, A.; Guan, F.; et al. Terrestrial laser scanning in forest inventories. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 63–77. [[CrossRef](#)]
6. Haag, M.; Nagel, H. Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Image Sequences. *Int. J. Comput. Vis.* **1999**, *35*, 295–319. [[CrossRef](#)]
7. Rother, D.; Sapiro, G. Seeing 3D objects in a single 2D image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009.
8. Nevatia, R.; Binford, T.O. Description and recognition of curved objects. *Artif. Intell.* **1977**, *8*, 77–98. [[CrossRef](#)]
9. Pentland, A.P. Perceptual organization and the representation of natural form. *Artif. Intell.* **1986**, *28*, 293–331. [[CrossRef](#)]
10. Huang, Q.; Wang, H.; Koltun, V. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.* **2015**, *34*, 1–10. [[CrossRef](#)]
11. Pound, P.M.; French, P.A.; Murchie, H.E.; Pridmore, T.P. Automated recovery of three-dimensional models of plant shoots from multiple color images. *Plant Physiol.* **2014**, *166*, 1688–1698. [[CrossRef](#)]
12. Kar, A.; Tulsiani, S.; Carreira, J.; Malik, J. Category-Specific Object Reconstruction from a Single Image. *arXiv* **2015**, arXiv:1411.6069.
13. Landy, M.; Movshon, J.A. Shape-from-X: Psychophysics and computation. In *Computational Models of Visual Processing*; MIT Press: Cambridge, MA, USA, 1991; pp. 305–330.

14. Bakshi, S.; Yang, Y. Shape from shading for non-Lambertian surfaces. In Proceedings of the International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994.
15. Zhang, R.; Tsai, P. Shape-from-shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 690–706. [CrossRef]
16. Todd, J.; Egan, E. The perception of shape from shading for Lambertian surfaces and range images. *J. Vis.* **2012**, *12*, 281. [CrossRef]
17. Torrance, K.E.; Sparrow, E.M. Theory for Off-Specular Reflection from Roughened Surfaces*. *J. Opt. Soc. Am. (JOSA)* **1967**, *57*, 1105–1114. [CrossRef]
18. Guan, L.; Franco, J.; Pollefeyns, M. 3D occlusion inference from silhouette cues. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007.
19. Didden, E.; Thorarinsdottir, T.; Lenkoski, A.; Schnörr, C. Shape from Texture Using Locally Scaled Point Processes. *Image Anal. Stereol.* **2015**, *34*, 161–170. [CrossRef]
20. Criminisi, A.; Zisserman, A. Shape from texture: Homogeneity revisited. In Proceedings of the British Machine Vision Conference (BMVC), Bristol, UK, 11–14 September 2000.
21. Verbin, D.; Zickler, T. Toward a universal model for shape from texture. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020.
22. Furukawa, Y.; Hernández, C. Multi-View Stereo: A Tutorial. *Now Found. Trends* **2015**, *9*, 1–148.
23. Parodi, P.; Piccioli, G. 3D Shape Reconstruction by Using Vanishing Points. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 211–217. [CrossRef]
24. Hamraz, H.; Jacobs, N.B.; Contreras, M.A.; Clark, C.H. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 219–230. [CrossRef]
25. Bo, Y.; Stefano, R.; Andrew, M.; Niki, T.; Hongkai, W. Dense 3D Object Reconstruction from a Single Depth View. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2820–2834.
26. Wu, J.; Zhang, C.; Xue, T.; Freeman, W.T.; Tenenbaum, J.B. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
27. Gadelha, M.; Maji, S.; Wang, R. 3D Shape Induction from 2D Views of Multiple Objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
28. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
29. Wu, J.; Wang, Y.; Xue, T.; Sun, X.; Freeman, B.; Tenenbaum, J. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In Proceedings of the International Conference on Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017.
30. Riegler, G.; Ulusoy, A.O.; Geiger, A. OctNet: Learning Deep 3D Representations at High Resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 27–30 June 2016.
31. Tatarchenko, M.; Richter, S.R.; Ranftl, R.; Li, Z.; Koltun, V.; Brox, T. What Do Single-view 3D Reconstruction Networks Learn? In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
32. Yuan, W.; Khot, T.; Held, D.; Mertz, C.; Hebert, M. PCN: Point Completion Network. In Proceedings of the International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018.
33. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. TopNet: Structural Point Cloud Decoder. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
34. Mescheder, L.; Oechsle, M.; Niemeyer, M. Occupancy Networks: Learning 3D Reconstruction in Function Space. In Proceedings of the Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
35. Wang, N.; Zhang, Y.; Li, Z. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
36. Zongji, W.; Feng, L. VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 2919–2930.
37. Xie, H.; Yao, H.; Zhou, S. GRNet: Gridding Residual Network for Dense Point Cloud Completion. In Proceedings of the European Conference on Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020.
38. Wen, X.; Xiang, P.; Han, Z.; Cao, Y.; Wan, P.; Zheng, W.; Liu, Y. PMP-Net++: Point Cloud Completion by Transformer-Enhanced Multi-step Point Moving Paths. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 852–867.
39. Zhou, H.; Cao, Y.; Chu, W.; Zhu, J.; Lu, T.; Tai, Y.; Wang, C. SeedFormer Patch Seeds Based Point Cloud Completion with Upsample Transformer. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022.
40. Li, S.; Gao, P.; Tan, X.; Wei, M. ProxyFormer: Proxy Alignment Assisted Point Cloud Completion with Missing Part Sensitive Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023.
41. Magistri, F.; Marks, E.; Nagulavancha, S.; Vizzo, I.; Läebe, T.; Behley, J. Contrastive 3D Shape Completion and Reconstruction for Agricultural Robots Using RGB-D Frames. *IEEE Robot. Autom. Lett.* **2022**, *7*, 10120–10127. [CrossRef]
42. Dai, A.; Qi, C.R.; Nießner, M. Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21 June–26 July 2017.

43. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
44. Wu, D.; Wang, Y.; Xia, S. Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. In Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 30 April 2020.
45. Zunair, H.; Hamza, A.B. Sharp U-Net: Depthwise convolutional network for biomedical image segmentation. *Comput. Biol. Med.* **2021**, *136*, 104699. [[CrossRef](#)] [[PubMed](#)]
46. Vinay, M.B.; Rekha, K.S. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *Int. J. Recent Technol. Eng. (IJRTE)* **2019**, *7*, 412–415.
47. Guo, M.; Cai, J.; Liu, Z.; Mu, T.; Martin, R.R.; Hu, S. PCT: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [[CrossRef](#)]
48. Fan, H.; Su, H.; Guibas, L.J. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 26 June–1 July 2016.
49. Diederik, P.; Kingma, J.B. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference for Learning Representations, San Diego, CA, USA, 14–16 April 2014.
50. Zhang, J.; Chen, X.; Cai, Z. Unsupervised 3D Shape Completion through GAN Inversion. In Proceedings of the Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.