Edge-constrained Point Cloud Segmentation for Plants

Jianzhong Zhu*, He Ren†, Kai Xie‡, Xue Jiang§ and Ruifang Zhai¶✉
College of Informatics, Huazhong Agricultural University, Wuhan, China
Email: *296753781@qq.com, †renhemail@foxmail.com, ‡xiekai_2021@163.com, §2252524406@qq.com
, ¶✉rfzhai@mail.hzau.edu.cn

Abstract—Point clouds is the powerful and practical data representation for plant parameterization and plant growth monitoring. The most critical step in the precise extraction of plant parameters is the accurate segmentation of plant point clouds, especially in the edge regions, as this directly affects the accuracy of the parameters. This research introduces an edge extraction algorithm to obtain edge points from the point cloud and utilizes these edge points to construct an edge module to enhance the Stratified Transformer model. The proposed method can better constrain the edge and learn the edge features of the point cloud. In this research, we compare our method with various methods on Pheno4D and S3DIS datasets. Our method shows significant improvement in edge segmentation, such as stem-leaf edges and window-wall edges.

Index Terms—Point cloud segmentation, Edge, Plant phenotype.

I. Introduction

In plant breeding, the selection of ideal plant traits is a key component in optimizing plant production [1] and in order to accelerate this selection, objective quantification of plant phenotypes is necessary [2]. In the past, this quantification of plant phenotypes relied on manual labor, which was time-consuming, expensive, and subject to large human error. With the rise of phenotypic analysis, various methods for extracting plant phenotypes from two-dimensional(2D) images have emerged, but 2D images are difficult to express the complex three-dimensional(3D) structure of plants with high quality due to the limitation of viewpoint. Therefore, the need to obtain the 3D morphology of plants becomes very important [3].

The 3D point cloud is the collection of numerous 3D coordinate points that effectively conveys the precise 3D shape and structure of an object or scene. In plant phenotype research, researchers use 3D scanners or multi-view reconstruction to capture or generate point cloud data of plants to obtain comprehensive information on the 3D morphological structure of plants, and then complete 3D phenotype analysis and growth monitoring.

However, the analysis of plant phenotypes using the whole plant point cloud enables the extraction of only macroscopic phenotypic parameters, such as height, width, and overall volume. To delve deeper into obtaining organ-level phenotypic parameters, it becomes imperative to perform precise point cloud segmentation based on plant organs. Conventional point cloud segmentation methods include edge-based, region-based, and model-fitting methods [4]–[6]. However, these methods often cannot achieve optimal results when directly applied to plant objects, and require special designs based on the features of plants.

In recent years, the rapid advancement of deep learning in the field of point clouds has opened up new possibilities for addressing this challenge. Deep learning models have the ability to automatically learn local and global features from point clouds, exhibiting strong feature extraction capabilities and generalization abilities. They can effectively handle challenges present in plant point clouds, such as noise, deformations, and occlusions.

Currently, there is relatively limited research and application of deep learning in plant point cloud segmentation. This is mainly due to the complex 3D structure of plants, which makes data acquisition and annotation very hard, and the lack of point cloud segmentation datasets specifically designed for plants. In addition, it is worth noting that many current models exhibit inadequate learning capability in terms of point cloud edge features, which leads to easy mis-segmentation of edges. Edges contain important information about the point cloud, which helps to improve the comprehension of the overall structure and shape of the point cloud.

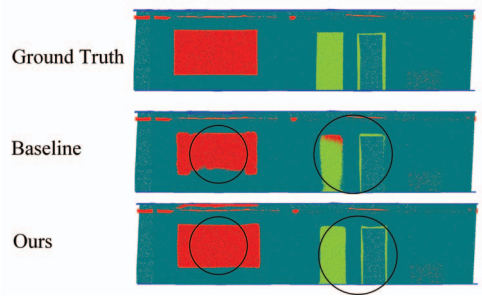

Fig. 1. It is a hallway in S3DIS Area 5. Our method outperforms the baseline ST in recognizing closed doors

Figure 1 illustrates the comparison between our proposed method and the baseline approach, showcasing an improvement in handling the divergence in edge regions. The contributions of this study are outlined as follows:

- We extract the edge points from the point cloud, employ them for data augmentation, and construct a large receptive field using the edge points and farthest

2023 China Automation Congress (CAC) | 979-8-3503-0375-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAC59555.2023.10450375

point sampling (FPS), which incorporates boundary information. This enables the network to better learn edge features and addresses the issue of segmentation errors in the edge regions.
- In order to mitigate the problem of imbalanced point distribution among different categories within the point cloud, we employed the weighted focal loss and assigned higher weights to edge points.
- We validated our method on the Pheno4D [7] and Stanford Large-Scale 3D Indoor Spaces(S3DIS) [8] datasets, and demonstrate that our method enhances the model's capability for edge segmentation.

II. Related Work

The methods for point cloud segmentation by deep learning are divided into two main categories: projection-based methods and point-based methods.

A. Projection-based methods

Projection-based methods in point cloud segmentation aim to project the unstructured point cloud data onto a regular space, such as 2D images, voxels, or high-dimensional grids. By projecting the point cloud onto these spaces, these methods enable the utilization of pre-existing research outcomes like Convolutional Neural Network (CNN). For example, MvCNN [9] projects point clouds into a set of multi-view 2D images and employs CNN to extract features, while VoxNet [10] voxelizes point clouds and utilizes 3D CNN for feature extraction. These approaches partially address the challenge of unstructured point clouds, but the projection process may result in some information loss.

B. Point-based methods

The PointNet series of networks, including PointNet [11] and PointNet++ [12], are the most prominent point-based methods. These methods employ multiple Multi-Layer Perceptron (MLP) to map the point cloud data into a high-dimensional feature space. Global max-pooling is then used to aggregate point-level features into global features of the entire point cloud. PointNet++ introduces hierarchical operations to process point cloud data, gradually extracting features from local to global levels, leading to more expressive feature representations and improved segmentation performance. Point Cloud Transformer (PCT) [13] shares a similar architecture with PointNet but incorporates the self-attention mechanism from Transformers. It uses neighbor embedding to encode local neighborhood information and utilizes it as input for the attention module, enabling PCT to learn spatial relationships between points. The feature concatenation of multiple attention modules allows PCT to capture both local and global contextual information when extracting point-level features.

Point Transformer [14] employs a ′vector self-attention′ operator to aggregate local features and a ′subtraction relation′ to generate attention weights. However, it lacks long-range context and exhibits limited robustness to various perturbations during testing. Stratified Transformer (ST) [15] addresses the limited receptive field issue in Point Transformer by incorporating hierarchical sampling. It combines locally dense points from the point cloud with sparsely sampled points using Farthest Point Sampling (FPS), overcoming the limitation of a limited receptive field. Nonetheless, the use of FPS in ST leads to significant loss of edge information in the sparsely sampled points, resulting in incomplete recognition of object edges and divergence issues.

Motivated by the capability of ST to enhance model segmentation performance through an enlarged receptive field, our work builds upon ST by incorporating an edge extraction algorithm to obtain the edges of the point cloud. We then combine these edge points with the sparsely sampled points obtained through FPS, thereby supplementing the expanded receptive field with crucial edge information. This approach aims to address the issue of divergence in point cloud edge segmentation and improve the segmentation results.

III. Our Methods

In the context of Transformer, the acquisition of larger perceptual fields often requires downsampling of point clouds, which poses the challenge of retaining information with a limited number of points. In this research, we focus on the edge information of the point cloud and address this challenge by combining edge extraction with the FPS method. This combination allows us to obtain a larger perceptual field with more informative content, enabling the network to better identify edges.

A. Overview

As shown in Figure 2a, our network employs an encoder-decoder architecture to handle point cloud data, where the input point cloud consists of XYZ coordinates and RGB color information. The input point cloud undergoes a series of preprocessing and data augmentation steps before being fed into the encoder. The encoder consists of point embedding modules for feature aggregation and is followed by multiple downsampling and transformer modules for feature encoding. Finally, the point cloud passes through the decoder, composed of multiple upsampling modules, to map the encoded features back to the original space of the point cloud, reconstructing the segmentation results.

As shown in Figure 2b, we perform voxelization downsampling on the input point cloud to obtain uniformly distributed points, and extract edge points. Then, the two types of points are concatenated and undergo data augmentation operations such as random rotation and adding random jitter to generate a series of new point clouds. As shown in Figure 2c, we construct a key consisting of edge points, FPS sampling points, and partial points, which provides edge information, large receptive

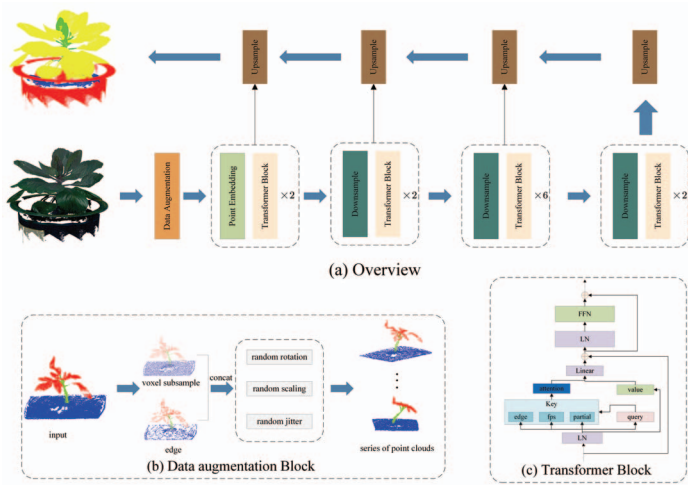

Fig. 2. (a) Overall Network Architecture;(b) Data Augmentation Block;(c) Composition of Transformer Modules .

field, and local information. This enables the network to have better receptive field and learning capabilities.

B. Edge Extraction Algorithm

The edge information of point clouds plays a crucial role in point cloud segmentation tasks. Within point clouds, edge points typically represent the geometric boundaries between objects or the interfaces separating different objects. These edge points carry essential features related to the shape and structure of objects, providing critical geometric information that aids in distinguishing and segmenting different objects or object components. By incorporating edge points as prior information into the learning process of the network, the robustness and generalization capabilities of point cloud segmentation models can be enhanced.

Algorithm 1 Edge Extraction Algorithm

Input: Point Cloud
Output: Edge points index

```
Find neighborhood and calculate D and G for each point
for each points p_i do
    Find potential edge points E_i within neighborhood
    if ∃p_k ∈ E_i such that D(p_i) < D(p_k) then
        D(p_i) = 0
    end if
end for
Obtain edge points Edge = {p_i | D(p_i) > Threshold}
```

In this research, we employed a neighborhood-based edge extraction algorithm for edge detection [16], which is primarily based on the distribution differences between geometric edge points and non-edge points in their neighborhood. Edge points tend to have a significant distance from the centroid of the neighborhood point set. By examining the gradient direction with the greatest change in distance, potential edges are identified, and a thresholding step is applied to select the final set of edge points, as shown in Figure 3.

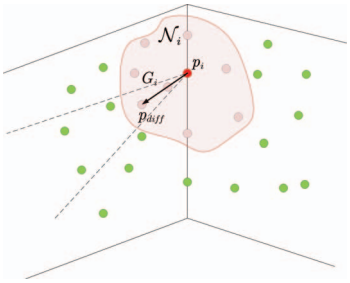

Fig. 3. The red point indicate the identified edge points, G_i indicates the gradient direction, and the points within the dashed line are the regions of potential edge points.

For a given point p_i in the point cloud, we first utilize a k-d tree to search for the set of k neighboring points denoted as $\mathcal{N}_i = \{p_j | j = 1, 2 \ldots, k\}$. Subsequently, we compute the centroid C_i of this neighborhood using Equation 1.

$$C_i = \frac{1}{k} \sum_{j=1}^{k} p_j \tag{1}$$

In Equation 1, p_j represents the neighboring points of p_i, and k denotes the number of neighboring points in the neighborhood.

$$D_i = \frac{|p_i - C_i|}{max(p_i, \mathcal{N}_i)} \tag{2}$$

To account for variations in local point cloud density, we normalize the distance between p_i and the centroid in Equation 2. In this equation, $max(p_i, \mathcal{N}_i)$ represents the maximum distance between the neighborhood points $\mathcal{N}_i$ and p_i. This normalization allows for a consistent comparison of distances regardless of the local point cloud density.

$$G_i = \frac{p_{diff} - p_i}{max(|D_i - D_j|)}, j = 1, 2 \ldots, k \tag{3}$$

The point $p_{diff} = \{p_j | max(|D_i - D_j|)\}$ in Equation 3 represents the point within the neighborhood $\mathcal{N}_i$ of p_i that exhibits the maximum difference in D values. This point is selected to determine the gradient of p_i, which points towards the neighborhood point with the highest difference. By considering this gradient, we can capture the direction of the most significant variation in the D values around p_i.

The points on the gradient direction G_i of point p_i within the neighborhood $\mathcal{N}_i$ are considered as potential edge points E_i, These potential edge points are depicted by the dashed line region in Figure 3. The set of potential edge

points E_i is defined as $\left\{P_i \mid \frac{(P_j - P_i)}{|P_j - P_i|} \cdot G_i > T\right\}$, The angle of the dashed line region is constrained by the threshold T.

C. Edge Receptive Field

Subsampling serves as a routine operation in diverse point cloud segmentation networks, facilitating the reduction of computational complexity and the equitable distribution of samples. It is noteworthy that with the increasing degree of subsampling, the intricate details of the point cloud, particularly the salient edge characteristics, progressively diminish in clarity. Our focus lies in addressing the challenge of preserving edge features during the subsampling process, enabling the network to effectively learn and segment these crucial edges for improved performance.

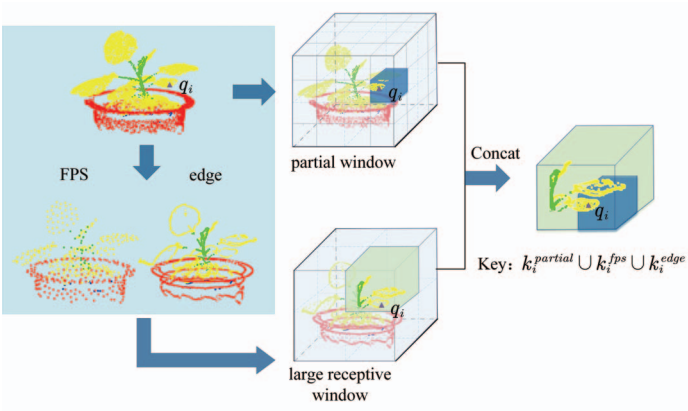

Fig. 4. The combination of edges and FPS constructs a larger perceptual field.

As depicted in Figure 4, we integrate the extracted edge points with the downsampled point cloud. By incorporating the edges seamlessly into the FPS process, we not only achieve a large receptive field but also ensure the inclusion of complete edge information within the attention window. This approach enables the attention mechanism to encompass more comprehensive edge details.

D. Boundary Focal Loss

In point cloud segmentation tasks, there may be a significant imbalance in the number of points belonging to different classes within the dataset. This class imbalance issue often negatively affects the training and performance of the models. To address this issue, we employed weighted focal loss as the loss function, as shown in Equation 4. We dynamically calculated the weights for each batch based on the point cloud quantities during the training process, giving higher weights to edge points.

$$\mathrm{FL}\left(p_t\right) = \sum -\alpha * h * \left(1 - p_t\right)^{\gamma} \log\left(p_t\right) \tag{4}$$

In the equation, α denotes the weight list for each class, h denotes the one-hot encoding, p_t denotes the output probability, and γ denotes the tuning parameter. $\alpha = 1/\log(1.1 + r)$, r is the proportion of each class in the batch.

IV. Experiments

In this section, we will introduce the datasets used in our study. Additionally, we will validate the improvement of the edge module in point cloud edge segmentation on the Pheno4D dataset and the S3DIS dataset [8].

A. Dataset

Pheno4D is an open dataset specifically designed for plant science research, aiming to provide rich 3D point cloud data for plant phenotype analysis and plant morphology studies. We utilized the annotated point clouds of 77 tomato plants from the Pheno4D dataset. Additionally, we also acquired point cloud data of multiple stages of cabbage using structured light devices, and used this dataset to validate the robustness of our proposed method.

S3DIS is a widely used indoor scene 3D point cloud dataset released by Stanford University. It consists of 271 rooms covering 6 different areas, and semantic labels are provided for these rooms, including a total of 13 categories. Following common dataset partitioning methods, we designated Area 5 as the testing set, while the remaining areas were used for training.

B. Experiments Setting

All experiments were conducted on a server equipped with 3 NVIDIA Tesla V100 GPUs. The performance evaluation was carried out on both the Pheno4D and S3DIS datasets.By conducting performance evaluations on these two datasets with different scales, we can verify the model's performance and robustness in various scenarios.

For the plant dataset, including the Pheno4D dataset and the Cabbage dataset, we adopted a training-to-testing ratio of 7:3. Initially, we randomly downsampled the point cloud to 100,000 points and set the neighborhood size to 400 for edge extraction. Regarding the extracted edges, we randomly sampled 3,000 points and concatenated them with the voxel-downsampled point cloud input to the network.

For the S3DIS dataset, according to the convention, we use Area 5 as the testset. Considering that larger scene scales lead to an increase in edge complexity, we randomly downsampled the point cloud to 300,000 points for edge extraction and randomly sampled 5,000 edge points.

C. Results

1) Plant: In the Pheno4D dataset and Cabbage dataset, we applied the edge module and compared it with the baseline results without using the edge module. As shown in Figure 5, it is obvious that with the use of the edge module, there is a significant improvement in the segmentation results at the boundaries between leaves and stems, as well as between leaves and pots. This shows that the edge feature can be better learned after using the edge

module, and the segmentation performance of the edge is improved.

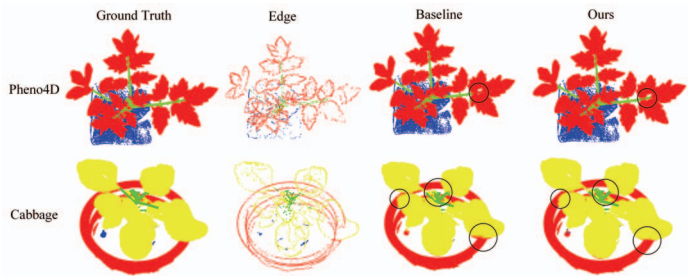

Fig. 5. Comparison of Results on Pheno4D Dataset and Cabbage Dataset.

TABLE I
Performance Comparison on Pheno4D and Cabbage Datasets

Dataset	Baseline	Ours	mIoU	mAcc	allAcc	soil	stem	leaf	pot
Pheno4D	✓		92.66	95.13	98.03	99.59	81.96	96.43	-
		✓	**93.6**	**95.89**	**98.5**	99.66	**84.02**	97.13	-
Cabbage	✓		90.98	94.13	**98.33**	92.68	76.4	97.94	96.88
		✓	**91.43**	**94.34**	98.09	92.06	**79.94**	97.55	96.18

We conducted a comprehensive comparison between our method and the baseline method, as shown in Table I, and our method achieved some improvements in the mIoU metrics, especially in the segmentation of stems. This result suggests that our method is more effective in capturing the boundary features between leaves and stems in plant point clouds. The visualization results in Figure 5 also confirm this conclusion.

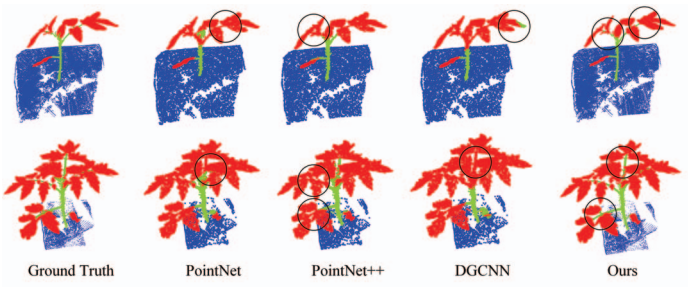

Fig. 6. Comparisons with Other Models on the Pheno4D Dataset.

TABLE II
Comparison of Results on Pheno4D

Model	mIoU	mAcc	allAcc	soil	stem	leaf
PointNet	81.25	85.98	95.32	96.98	56.52	90.25
PointNet++	87.47	92.03	97.18	99.26	69.54	93.59
DGCNN	72.84	79.04	93.23	98.09	35.48	84.94
Ours	**93.6**	**95.89**	**98.5**	**99.66**	**84.02**	**97.13**

In addition, we compared multiple models on the Pheno4D dataset, and the results show that our method performs the best, as shown in Table II. PointNet only performs global pooling to aggregate features, which leads to a severe lack of local information. PointNet++ splits the data into multiple small regions to extract features, which significantly improves performance. In contrast, the DGCNN method of extracting features using constructed graphs does not seem to be applicable to situations involved in the intercrossing of plant stems&leaves. Our method, based on Transformer and edge techniques, achieves the clearest segmentation of leaves and stems, as shown in Figure 6.

2) S3DIS: According to the convention, we evaluated the performance in Area 5 and compared it with other methods. As shown in Figure 7, in the edge, we can observe the edges extracted by the edge extraction algorithm, including walls, clutter, tables etc. This allows the model to have a better perception of relative positions and enhances its learning ability for edges. Additionally, with the use of the edge module, the scattered errors around the walls are significantly reduced, and the missegmentation of clutter on the indoor table is also significantly reduced.

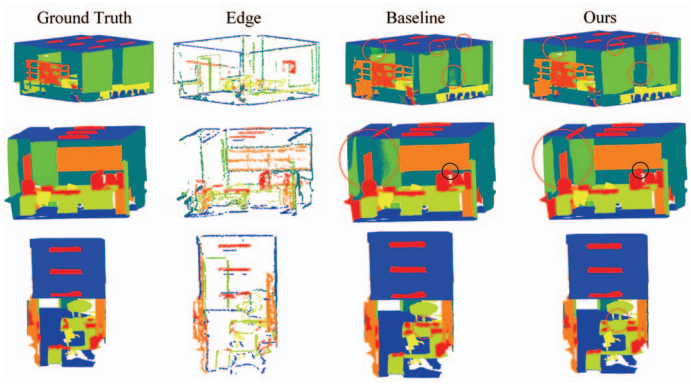

Fig. 7. Comparison of Results on S3DIS. The first row shows the outdoor view of a conference room. The second and third rows display indoor scenes of two offices.

TABLE III
Performance Comparison: The best results highlighted in bold and those surpassing the baseline marked with an asterisk. Compared to other methods, our method demonstrates strong segmentation capabilities for objects with geometric boundaries such as windows and boards.

methods	mIoU	OA	mACC	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [11]	41.1	-	49	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	59	52.6	5.9	40.3	26.4	33.2
SegCloud [17]	48.9	-	57.4	90.1	96.1	69.9	0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13	41.6
PointCNN [18]	57.3	85.9	63.9	92.3	98.2	79.4	0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph [19]	58	86.4	66.5	89.4	96.9	78.1	0	42.8	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
PCT [13]	61.3	-	67.7	92.5	98.4	80.6	0	19.4	61.6	48	76.6	85.2	46.2	67.7	67.9	52.3
KPConv [20]	67.1	-	72.8	92.8	97.3	82.4	0	23.9	58	69	81.5	91	75.4	75.3	66.7	58.9
JSENet [21]	67.7	-	-	93.8	97	83	0	23.2	61.3	71.6	89.9	79.8	75.6	72.3	72.7	60.4
CGA-Net [22]	68.6	-	-	94.5	98.3	83	0	25.3	59.6	71	**92.2**	82.6	76.4	77.7	69.5	61.5
ST [15]	70.96	90.77	76.97	94.83	98.67	82.91	0	39.64	55.69	**79.92**	84.1	**92.84**	**83.09**	78.8	68.42	**63.54**
Ours	**71.9***	**91.51***	**78.27***	**95.3***	**98.69***	**86.48***	0	**47.07***	**65.29***	73.3	85.84*	90.55	70.86	**78.81***	**80.29***	62.87

As shown in Table III, the addition of the edge module resulted in a significant improvement in segments with clear geometric edges as boundaries, such as walls, boards, and especially windows. However, there was no observed improvement for doors and chairs. This is because the 5000 edge points provide a relatively sparse representation when depicting an entire room, and the sparsity of the edge points becomes more pronounced for smaller objects such as doors and chairs. As a result, the ability of the edge points to guide the model on these objects is compromised.

D. Ablation Study

We conducted ablation experiments on the widely used S3DIS dataset. In order to reflect the differences in the learning capabilities of the network and accelerate training, we reduced the number of samples used for data augmentation in the ablation experiments, which resulted in a slight performance decrease.As shown in Table IV, the addition of the edge module led to improvements in mIoU, OA, and mACC to varying degrees. Furthermore, the inclusion of the boundary focal loss, which balanced the samples and constrained the edges, resulted in a significant improvement in mACC.

TABLE IV
The performance evaluation of ablation networks on the S3DIS Area 5

edge	boundary focal loss	mIoU	OA	mACC
		70.38	91.48	76.59
✓		70.64	91.59	76.91
✓	✓	70.73	91.05	78.66

V. Conclusion

In this paper, we employ a region-based edge extraction algorithm to obtain the edges of the point cloud. Based on this, we propose a universal edge module that uses edges to augment the downsampled point cloud, and combines edge points with FPS to create a large receptive field with boundary information. Additionally, to address the issue of point imbalance and constrain the edges, we utilize boundary focal loss. With the above methods, we have improved the model's capability of segmentation of edges. The edge extraction algorithm extracts regular geometric edges well and falls short for irregular edges, which leads to the model's inability to improve on all edge segmentation. Finally, we evaluate the performance of our approach on the Pheno4D dataset and the S3DIS dataset. The results indicate that our method enhances the network's ability to learn boundary features and improves its segmentation capability for edges.

References

[1] J. L. Araus and J. E. Cairns, "Field high-throughput phenotyping: the new crop breeding frontier," Trends in plant science, vol. 19, no. 1, pp. 52–61, 2014.

[2] Y. Lin, "Lidar: An important tool for next-generation phenotyping technology of high potential for plant phenomics?" Computers and electronics in Agriculture, vol. 119, pp. 61–73, 2015.

[3] S. Dhondt, N. Wuyts, and D. Inzé, "Cell to whole-plant phenotyping: the best is yet to come," Trends in plant science, vol. 18, no. 8, pp. 428–439, 2013.

[4] G. Vosselman, B. G. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," International archives of photogrammetry, remote sensing and spatial information sciences, vol. 46, no. 8, pp. 33–38, 2004.

[5] T. Rabbani, F. Van Den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," International archives of photogrammetry, remote sensing and spatial information sciences, vol. 36, no. 5, pp. 248–253, 2006.

[6] A. Jagannathan and E. L. Miller, "Three-dimensional surface mesh segmentation using curvedness-based region growing approach," IEEE Transactions on pattern analysis and machine intelligence, vol. 29, no. 12, pp. 2195–2204, 2007.

[7] D. Schunck, F. Magistri, R. A. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann et al., "Pheno4d: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis," Plos one, vol. 16, no. 8, p. e0256340, 2021.

[8] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1534–1543.

[9] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 945–953.

[10] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2015, pp. 922–928.

[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in neural information processing systems, vol. 30, 2017.

[13] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," Computational Visual Media, vol. 7, pp. 187–199, 2021.

[14] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 16 259–16 268.

[15] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8500–8509.

[16] D. Bazazian, J. R. Casas, and J. Ruiz-Hidalgo, "Fast and robust edge extraction in unorganized point clouds," in 2015 international conference on digital image computing: techniques and applications (DICTA). IEEE, 2015, pp. 1–8.

[17] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in 2017 international conference on 3D vision (3DV). IEEE, 2017, pp. 537–547.

[18] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," Advances in neural information processing systems, vol. 31, 2018.

[19] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4558–4567.

[20] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6411–6420.

[21] Z. Hu, M. Zhen, X. Bai, H. Fu, and C.-l. Tai, "Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16. Springer, 2020, pp. 222–239.

[22] D. Marin, Z. He, P. Vajda, P. Chatterjee, S. Tsai, F. Yang, and Y. Boykov, "Efficient segmentation: Learning downsampling near semantic boundaries," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2131–2141.