remote sensing

MDPI

Article

Deep Ground Filtering of Large-Scale ALS Point Clouds via Iterative Sequential Ground Prediction

Hengming Dai [1], Xiangyun Hu [1,2,3,*], Zhen Shu [1], Nannan Qin [4] and Jinming Zhang [5,6]

[1] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[2] Hubei Luojia Laboratory, Wuhan 430079, China
[3] Institute of Artificial Intelligence in Geomatics, Wuhan University, Wuhan 430079, China
[4] School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China
[5] Key Laboratory of Network Information System Technology, Institute of Electronic, Chinese Academy of Sciences, Beijing 100190, China
[6] The Aerospace Information Research Institute, Chinese Academic of Sciences, Beijing 100190, China
* Correspondence: huxy@whu.edu.cn; Tel.: +86-27-6877-1528; Fax: +86-27-6877-8086

Abstract: Ground filtering (GF) is a fundamental step for airborne laser scanning (ALS) data processing. The advent of deep learning techniques provides new solutions to this problem. Existing deep-learning-based methods utilize a segmentation or classification framework to extract ground/non-ground points, which suffers from a dilemma in keeping high spatial resolution while acquiring rich contextual information when dealing with large-scale ALS data due to the computing resource limits. To this end, we propose SeqGP, a novel deep-learning-based GF pipeline that explicitly converts the GF task into an iterative sequential ground prediction (SeqGP) problem using points-profiles. The proposed SeqGP utilizes deep reinforcement learning (DRL) to optimize the prediction sequence and retrieve the bare terrain gradually. The 3D sparse convolution is integrated with the SeqGP strategy to generate high-precision classification results with memory efficiency. Extensive experiments on two challenging test sets demonstrate the state-of-the-art filtering performance and universality of the proposed method in dealing with large-scale ALS data.

Keywords: airborne laser scanning; ground filtering; deep reinforcement learning; sparse convolutional neural network

check for updates

Citation: Dai, H.; Hu, X.; Shu, Z.; Qin, N.; Zhang, J. Deep Ground Filtering of Large-Scale ALS Point Clouds via Iterative Sequential Ground Prediction. *Remote Sens.* **2023**, *15*, 961. https://doi.org/10.3390/rs15040961

Academic Editor: Giuseppe Casula

Received: 25 December 2022
Revised: 4 February 2023
Accepted: 6 February 2023
Published: 9 February 2023

1. Introduction

High-quality Digital Elevation Model (DEM) generation is a prerequisite for a variety of environmental applications, including forest wildfire fuel consumption estimation [1], forest inventory [2], archaeological surveying [3], landslide detection [4], and so on. Airborne Laser Scanning (ALS) has unique advantages in producing high-quality DEM, accounting for its penetration capability and efficiency in acquiring high-density large-scale point clouds with complex terrain details. The critical step to generating DEM from ALS data is separating point clouds into ground and non-ground points, often called ground filtering (GF). Nevertheless, such a task remains challenging due to the variations in the geometric structure in both terrain surface and multitudinous land covers [5]. The vast intra-class variance and inter-class similarities make it overly difficult to distinguish between ground points and non-ground points accurately [6].

Traditional GF methods are based on geometric rules and can be roughly categorized into slope-based, morphology-based, and surface-based methods. The slope-based methods [7,8] analyze the slope value in a local context and distinguish non-ground points by setting a threshold. The morphology-based methods [9,10] apply the mathematical morphology operation accordingly to remove non-ground points. The surface-based methods progressively select points from the raw point clouds to fit a ground surface, which can be

achieved by the Triangulated Irregular Network (TIN) [11] or interpolation [12,13]. The Cloth Simulation Filter (CSF) [14] is another representative GF method in recent years, which utilizes a physical procedure to simulate a virtual cloth deforming into a ground surface. While performing well in specific scenarios, these rule-based methods may produce unsatisfactory results in complex scenes [15] and require experiential parameter-tuning [16].

To increase the robustness and automation level, many classical machine learning algorithms have been introduced to the point cloud classification problem, such as Support Vector Machine (SVM) [17], random forest (RF) [18], and Conditional Random Field (CRF) [19]. Kang et al. [20] propose combining the geometric features calculated from the point cloud and the spectral features from images and conducting point cloud classification by a Bayesian network. Zhang et al. [21] propose classifying point clouds of urban areas by utilizing a support vector machine. In the meantime, several studies use CRF to mine spatial contextual information and achieve good classification results [22,23]. Niemeyer et al. [24] combines the RF with CRF to conduct point cloud classification. However, the classical machine learning classifiers mainly rely on hand-crafted features, which may lack the generalization and representation ability [16].

In the past few years, deep learning methods have been thriving in the context of point cloud processing [25]. Depending on the data representation of the network's inputs, these processing pipelines can be roughly grouped into three categories, namely, image-based, voxel-based, and point-based methods. Image-based methods map points into images and utilize proven 2D convolutional neural networks to conduct classification [26] or segmentation [27]. In early investigations on tackling GF problems with deep learning, Hu and Yuan [28] propose to project each point into a three-channel image by calculating the difference of elevation in a local context, then apply a 2D CNN to classify ground and non-ground points. Rizaldy et al. [29] calculate the pixel value based on the lowest or highest point within each cell and apply a Fully Convolution Network (FCN) to conduct ground filtering or multi-class classification. Yang et al. [30] brings up a new way to map a point into an image by calculating geometric features. Wang et al. [31] utilize the multi-scale strategy and attention mechanism to improve the classification performance. These image-based methods are effective in many situations but also suffer from inherent information loss to geometric structures during 3D–2D rendering [32,33] and may be problematic in forest areas [34]. Point-based methods process unordered point clouds directly, thus the original geometric structure is preserved. Pioneered by PointNet [35] and PointNet++ [36], numerous studies have been putting effort into point-based methods recently and various networks emerged. PointCNN [37] learns a transformation that benefits the weighting of point features and the permutation of points into canonical order. Wang et al. [38] use a dynamic graph for feature aggregation, which fully explores the local and global information. The KPConv [39] uses a point-based convolution kernel for feature learning and achieved great performance. Hu et al. [40] bring up a lightweight feature aggregation module and use random sampling to improve efficiency, which makes it possible to deal with outdoor point clouds on a large scale. Janssens-Coron and Guilbert [41] made a preliminary attempt to conduct ground filtering based on PointNet [35]. Jin et al. [32] propose a point-based FCN for ground filtering. Zhang et al. [34] propose a novel TinEdgeConv for ground filtering in forest areas. Li et al. [33] utilize the KPConv operator with the self-attention mechanism for ground filtering of Unmanned Aerial Vehicles (UAV) point clouds. Fareed et al. [42] use PointCNN [37] to tackle the ground filtering of UAV point clouds in agricultural fields and prove that the PointCNN is superior to several frequently used GF algorithms in classification accuracy and transferability. Currently, the point-based methods have made a lot of progress in the ground filtering of mountain areas [9,32,33]; however, their application in a hybrid scenario with a large scale is restricted by the sampling extent since the number of input points is limited. This situation limits the generality of such methods because they may not correctly handle the large-scale object in urban areas [6,43,44]. Voxel-based methods regularize point clouds with a 3D grid, which is the 3D counterpart of pixels. This makes it possible to extend the full-fledged

2D convolutional neural networks into 3D space, but the computational expense grows exponentially at the same time. Yotsumata et al. [45] use voxel representation on each point with its local context and conduct ground filtering by classifying each voxelized input using a 3D CNN. However, the dense voxel representation cannot capture large semantic context with fine details due to the huge memory burden. Fortunately, the progress in sparse convolution [46–49] may alleviate this problem by exploiting the intrinsic sparsity of point cloud data. Schmohl and Sörgel [50] use submanifold convolutional networks [48] for the semantic segmentation task of ALS data, but a high voxel resolution still limited the sampling extent of spatial context. The sparse convolutional network has been widely applied to many 3D vision tasks and achieved great performance, including but not limited to indoor scene recognition [51], 3D object detection [52,53], semantic and instance segmentation [53,54]. However, its application in large-scale ALS point cloud processing is relatively rare to our best knowledge. Similar to the point-based method, the most important reason is that the huge amount of data limits either the spatial resolution or contextual information[32]. To sum up, the recent deep-learning-based GF methods perform well in specific scenes but lack universality when dealing with a complex situation because of the conflict between the spatial resolution and range of context. How to keep high enough spatial resolution for distinguishing terrain details and near-ground points while obtaining a large context for identifying large-scale objects is still a question to answer.

We noticed that in the GF problem, the bare terrain is a two-dimensional manifold embedded into 3D space. In this case, the ground points located on the terrain surface share similar geometric structures in a local range, and the classification results of the adjacent points are highly correlated, which indicates that the ground points can be retrieved gradually in a sequential manner. Based on this observation, we formulate the GF problem into a sequential prediction task using points-profiles, retrieving the ground points in each points-profile iteratively. In summary, we propose a novel GF pipeline named sequential ground prediction (SeqGP), which can acquire high spatial resolution and large contextual information simultaneously. The main contributions of this paper are as follows:

- A novel deep-learning-based GF pipeline is proposed by converting the GF problem into a sequential ground prediction task based on points-profiles, which keeps high spatial resolution while acquiring a large context.
- An HCF module is proposed to capture large-scale contextual information efficiently and facilitate the recognition of large-scale artificial objects.
- The extensive experiments demonstrate that the SeqGP achieves state-of-the-art GF performance and universality in dealing with large-scale objects and mountain areas simultaneously.

The remainder of this paper is organized as follows. Section 2 introduces the materials used in this study and presents the formulation of the SeqGP in detail. Section 3 presents the comprehensive experimental results and analysis to demonstrate the effectiveness of the proposed method. In Section 4, we discuss certain advantages and limitations of the proposed method and present several aspects for future research. Finally, Section 5 draws the conclusions.

2. Materials and Methods

In this section, we introduce the formulation of the proposed GF pipeline. First, the description of the datasets used in our study and the techniques correlated to the proposed method is presented in Section 2.1. The SeqGP is described in Section 2.2. Finally, the Sections 2.3 and 2.4 present the implementation details and the evaluation metrics, respectively.

2.1. Materials

2.1.1. Datasets

OpenGF. OpenGF (https://github.com/Nathan-UW/OpenGF, accessed on 15 April 2021) is the first public large-scale GF dataset [6]. The dataset collects diverse terrain scenes

from 4 countries and covers approximately 47.7 km^2 with more than 542 million points. The test set contains hybrid terrain scenes with various land covers, which are quite challenging for existing deep-learning-based GF algorithms. Test Site I covers about 6.6 km^2, which contains villages, small cities, and mountains. Test Site II is a metropolitan area covering about 1.1 km^2, which contains a variety of non-ground objects with large-scale variation. The two test sites are illustrated in Figure 1, some statistical information can be seen in Table 1.

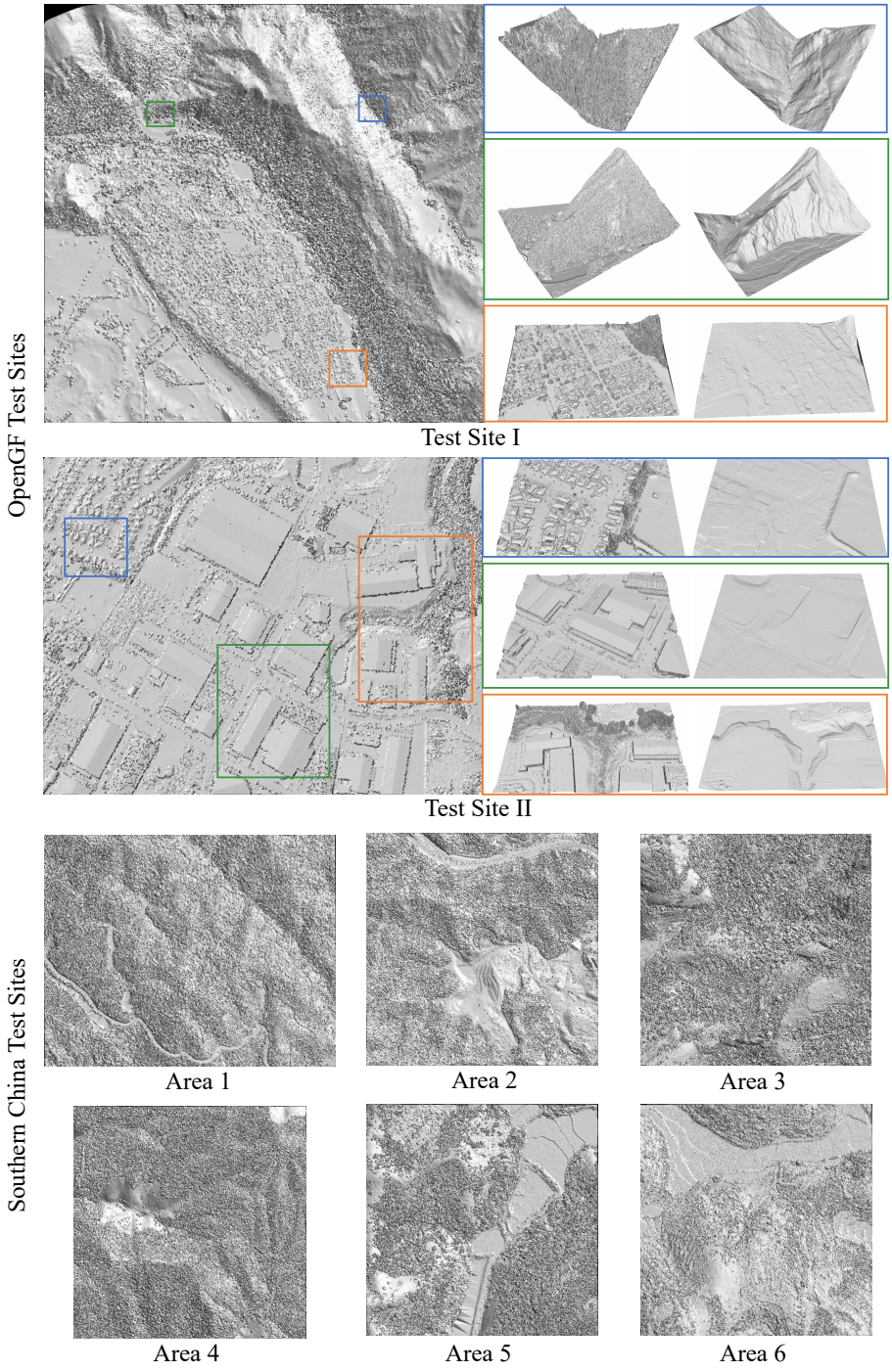

Figure 1. The test sites of OpenGF dataset and Southern China dataset.

Southern China dataset. We also utilize a challenging testing set introduced by Zhang et al. [34] for further analysis of the generalization ability of the proposed method. The testing set contains six areas collected from southern China with different terrain conditions. The average points density of six areas ranged from 0.7 points/m^2 to 37 points/m^2.

The six test areas are illustrated in Figure 1, some statistical information can be seen in Table 1.

We use the public OpenGF dataset [6] to demonstrate the effectiveness of our method to filter out large buildings while keeping terrain details in mountain areas. The experiment on the Southern China dataset [34] shows the good generalization ability of the proposed method.

Table 1. Statistical information about the test sites of OpenGF dataset and Southern China dataset.

	OpenGF		**Southern China**					
	Test Site I	**Test Site II**	**Area 1**	**Area 2**	**Area 3**	**Area 4**	**Area 5**	**Area 6**
Area (km^2)	6.60	1.10	0.80	1.00	0.07	1.00	0.07	0.06
Number of points (M)	46.00	6.22	0.62	3.30	2.53	3.24	2.57	1.25
Density ($points/m^2$)	6.97	5.65	0.78	3.30	36.24	3.24	36.71	20.8

2.1.2. Relevant Concepts

In this section, we introduce some concepts relevant to the proposed method and provide a review of them briefly.

Profile-based point cloud analysis. The profile-based partitioning is computationally efficient and has been widely used in point cloud processing. The profile has the ability to represent manifold-like structures including but not limited to power-line [55], curves [56], and tunnels [57], as well as the bare terrain [58]. Inspired by this, we proposed to incorporate the profile representation with deep learning to alleviate the conflict between high spatial resolution and large context for the existing deep-learning-based GF method, which has not been investigated before. In the meantime, the correlation between the adjacent profiles is learned by the neural network in the proposed SeqGP, rather than explicitly constructed by geometric rules or statistical analysis.

Deep Reinforcement Learning. The core problem that Reinforcement Learning (RL) deals with is sequential decision-making, which has two major characteristics. On the one hand, the adjacent decisions are highly correlated and new decisions are made based on previous ones. On the other hand, the RL agent aims to maximize the total rewards, which accumulate until the last decision is made.

DRL is introduced by deep Q-network (DQN) [59], which integrated neural networks with Q-learning [60]. A growing number of DRL algorithms have been proposed and many of them are applied to 3D computer vision tasks. 3DCNN-DQN-RNN [61] utilizes DQN for eye window localization and improved the efficiency of indoor point cloud parsing. IteR-MRL [62] investigates a multi-agent RL framework for 3D medical image segmentation by interactively refining the segmentation probability. RL-GAN-Net [63] tackles the problem of the point cloud shape completion by applying an RL agent trained by Deep Deterministic Policy Gradient (DDPG) [64] to manipulate the latent vector of the Generative Adversarial Network (GAN) with continuous action. Although demonstrating its practicality in many 3D vision tasks, applying DRL to GF tasks is not yet been fully explored.

In fact, there are several ways to tackle sequence labeling tasks, including but not limited to using the LSTM-based recurrent network [65], Transformer [66], and DRL [67]. All of these methods apply to our task. Considering that the iterative retrieval of bare terrain mainly relies on modeling the relationship between the adjacent points-profiles, while the RL method has the characteristics of exploration and exploitation, involving DRL may lead the network to find more implicit relations between them. Furthermore, the training data of DRL are state-action pairs stored in the replay memory rather than the whole sequence [59], which has two major benefits for our implementation. First, the dependencies of the training data are decoupled and thus helping the network on learning the transition between the adjacent states. Second, the input state is represented by a profile-stack, thus alleviating the memory burden and allowing high-resolution voxels.

Based on the above considerations, we employ DRL algorithms in this paper to tackle the sequence labeling of ground points from ALS point clouds.

2.2. Methods

The proposed method tackles the GF problem by labeling the points-profile iteratively. There are two main advantages of this formulation. First, the data volume of the network's inputs is reduced significantly, thus guaranteeing a large semantic context while satisfying high spatial resolution under limited GPU memory. Second, the prediction of each points-profile is considered in subsequent steps, thus preserving the continuity of the bare terrain to some extent. To implement the above idea, we partition the point clouds into a sequence of points-profiles. Afterward, each time the current points-profile and the previous ground information are fed into a semantic segmentation model to classify the ground points in the current step. Since we use voxel representation in our method, the large spatial context with high voxel resolution leads to a large spatial size of the input voxels. We further propose a High-Level Context Fusion (HCF) module to increase the receptive field and incorporate large-scale contextual information. In the meantime, the sequential prediction procedure is considered a Markov Decision Process (MDP) and optimized by a Deep Reinforcement Learning (DRL) framework. The pipeline of the proposed method can be seen in Figure 2.

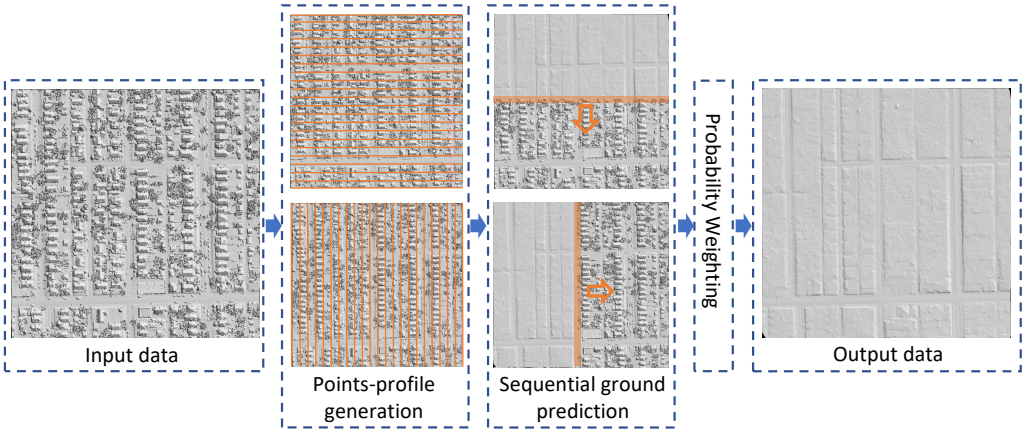

Figure 2. Overview of the proposed framework for the digital elevation model (DEM) extraction from large-scale airborne laser scanning (ALS) point cloud (The TIN-based DSM and DEM are used for visualization).

2.2.1. Points-Profiles Generation

The input point clouds are first partitioned into many points-profiles along a certain horizontal direction, in which the thickness of each profile is controlled by a hyper-parameter d. We illustrate the details of points-profiles generation by taking the x-axis as an example. Given a point set $P = \{p_1, p_2 \ldots, p_n\}$ with $p_i = (x_i, y_i, z_i)$, points in P are split into profiles by their spatial coordinates in x-axis. Suppose that the points in P are located in the range $[x_{min}, x_{max}]$ in the x-axis. Then, the point p_i is assigned to the k_{th} slice, where $k = floor((x_i - x_{min})/d)$ and x_i is p_i's coordinate in the x-axis. Eventually, N slices are generated in total, where $N = ceil((x_{max} - x_{min})/d)$. The $floor$ and $ceil$ represent the round down and round up operation, respectively. After the slicing process, all points in P are grouped into N slices, wherein each slice also forms a point set χ_i, then the input point clouds are divided into N ordered sets of points $X = \{\chi_1, \chi_2, \ldots, \chi_N\}$, where χ_i denotes the set of points that belong to i_{th} slice, and the minimum x coordinate value in χ_1 equal x_{min}, the maximum x coordinate value in χ_N equal x_{max}. We define χ_i as a points-profile, the illustration of the points-profile can be seen in Figure 3.

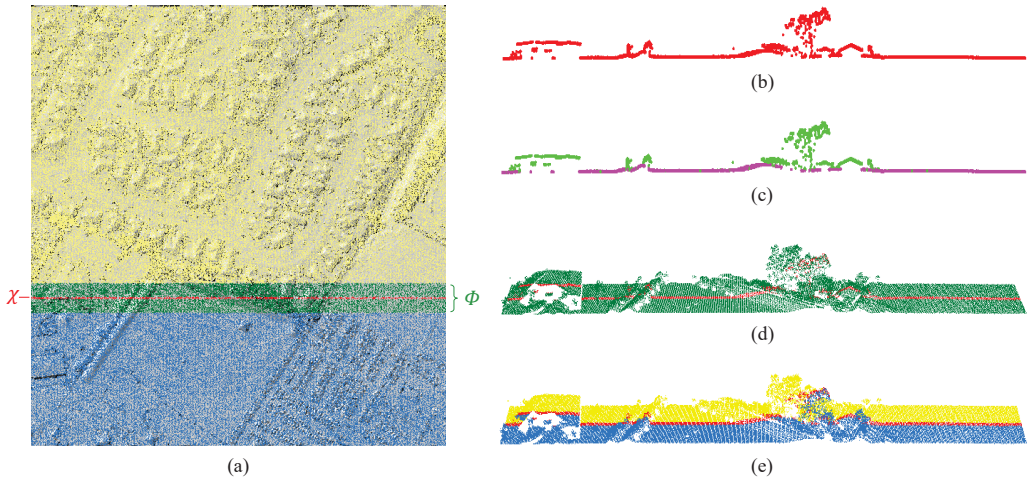

Figure 3. Illustration of Points-profile and Profile-stack: (**a**) Points-profile and Profile-stack in patch data (with DSM), where blue points denote areas that have been classified and yellow points denote areas that have not been classified, green points denote profile-stack and red points denote points-profile; (**b**) Points-profile; (**c**) Ground truth in points-profile; (**d**) Profile-stack; (**e**) Profile-stack containing both classified and unclassified areas.

2.2.2. Sequential Ground Prediction

After the points-profiles set X is generated, our goal is to retrieve every ground point in each points-profile from χ_1 to χ_N. The overview of the sequential ground prediction (SeqGP) can be seen in Figure 4. Each time, the semantic segmentation model observes the previous ground information and classifies the ground points in the current points-profile. The agent aims to find a prediction sequence $(pred_1, pred_2, \ldots, pred_N)$, in which each prediction $pred_i$ assigns semantic labels to every point in χ_i. Once χ_i is labeled, the classification results are applied to the environment and the agent makes the next prediction of χ_{i+1} based on a new observation of the environment. This procedure is formulated as a Markov decision process composed of a state space S, an action space A, and a reward function R. Detailed descriptions are given next.

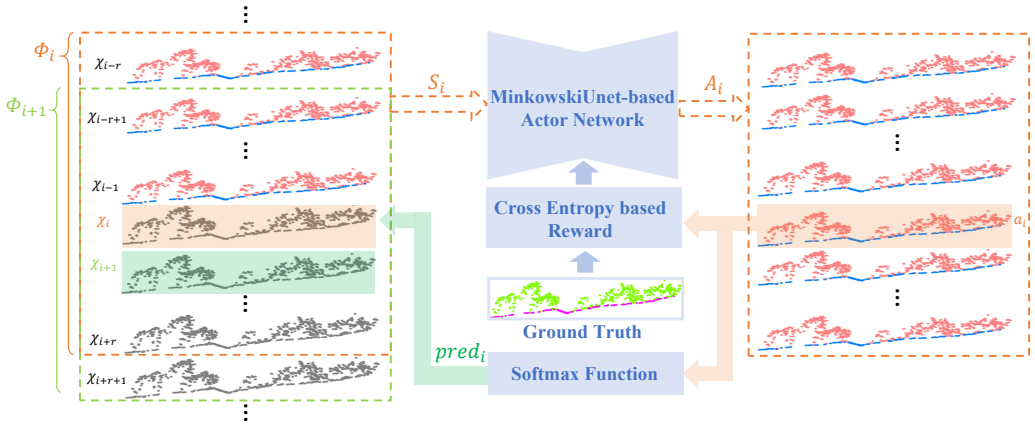

Figure 4. Overview of sequential ground prediction based on deep reinforcement learning. The S, A, χ, and ϕ denote the state, action, points-profile, and profile-stack, respectively.

State. The state carries the information about the environment that the agent can acquire. We define a profile stack Φ_i that includes profile χ_i and its adjacent profiles, $\Phi_i = [\chi_{i-r}, \chi_{i-r+1}, \ldots, \chi_i, \ldots, \chi_{i+r-1}, \chi_{i+r}]$, where r controls the number of profiles the agent can observe in each time step as shown in Figure 4. At time step t, the state S_i^t is defined as a sparse tensor generated by all points in Φ_i with their corresponding feature vector $f = [1, g]$, where g is a ground point indicator that indicates whether a point has been labeled as a ground point or not by previous predictions. We set 1 for an extra dimension

of the points' feature vector because the input of the Minkowski network is defined to be a non-zero vector. For the points that have been classified by the agent, g is obtained by applying the softmax function to the agent's previous actions, where the other unclassified points are initialized with $g = -1$. The definition of classified and unclassified areas are as shown in Figure 3e.

$$g = \begin{cases} -1 & Unclassified \\ softmax(a) & Classified \end{cases} \tag{1}$$

Hence, the input state encodes the retrieved ground surface information along with the geometric structures of the original point cloud. This formulation enables the agent to act based on previous predictions. The illustration of the points-profile and profile-stack are shown in Figure 3.

Action. The agent acts similarly to a conventional segmentation network and outputs segmentation probability. At time step t, the action $A_i^{(t)}$ is defined as follows:

$$A_i^{(t)} = [a_{i-r}^{(t)}, a_{i-r+1}^{(t)}, \dots a_i^{(t)}, \dots, a_{i+r-1}^{(t)}, a_{i+r}^{(t)}] \tag{2}$$

which gives a segmentation probability to each profile in Φ_i.

Reward. The reward indicates how much profit the agent can receive by taking a certain action. Because we aim to obtain a classification result at each time step t, we utilize cross-entropy and subtract it as the reward. y_i denotes the ground truth of χ_i.

$$R_i^{(t)} = y_i log(a_i^{(t)}) + (1 - y_i) log(1 - a_i^{(t)}) \tag{3}$$

During the inference stage, the binary segmentation result on each profile χ_i is obtained sequentially as described in Algorithm 1.

Algorithm 1 Sequential ground prediction algorithm

Input: Points-profile set $X = \{\chi_1, \chi_2, ..., \chi_N\}$
Output: prediction sequence $(pred_1, pred_2, ..., pred_N)$

```
Initialize i = 1
while i <= N do
    generate state S_i with Φ_i and all points' f
    obtain A_i
    obtain pred_i = softmax(a_i)                ▷ a_i ∈ A_i
    update points feature vector f in χ_i
    i++
return (pred_1, pred_2, ..., pred_N)
```

2.2.3. Training

To perform sequential prediction and retrieve the bare terrain gradually, we utilize the DDPG algorithm, in which an actor network and a critic network are present. The actor network learns a mapping from state S to action A, which is named policy $\pi(S)$. The critic $Q(S, A)$ estimates the amount of reward that the agent may get when adopting a policy. The critic is optimized by the Bellman equation [60] and the replay memory is used for random sampling of training data:

$$Q(S_i^{(t)}, A_i^{(t)}) = R_i^{(t)} + \gamma Q(S_{i+1}^{(t+1)}, \pi^{(t+1)}) \tag{4}$$

Here, $R_i^{(t)}$ is a reward that the agent may gain when taking a certain action $A_i^{(t)}$ based on state $S_i^{(t)}$ at time step t. Based on the points-profile formulation, we also have:

$$Q_i^{(t)} = [q_{i-r}^{(t)}, q_{i-r+1}^{(t)}, \dots q_i^{(t)}, \dots, q_{i+r-1}^{(t)}, q_{i+r}^{(t)}] \tag{5}$$

By maximizing the critic's estimation of Q, the actor π is optimized. Specifically, the actor would decide which points belong to the ground category in each points-profile and the critic gives a prediction on the reward. As mentioned in Section 2.2.2, the action $a_i^{(t)} \in A_i^{(t)}$ at state $S_i^{(t)}$ provides the prediction result on $\chi_i \in \Phi_i$. In the next step, we also have $\chi_i \in \Phi_{i+1}$ based on the formulation of the profile-stack. Thus, the action $A_{i+1}^{(t+1)}$ at time step $t+1$ contains $a_i^{(t+1)}$ which gives a segmentation probability of χ_i at time step $t+1$. During the one-step look-ahead training process in RL, $Q^{(t+1)}$ is obtained by

$$Q(S_{i+1}^{(t+1)}, \pi^{(t+1)}) = q_i^{(t+1)} + R_{i+1}^{(t+1)} \tag{6}$$

where $q_i^{(t+1)} \in Q_{i+1}^{(t+1)}$ and $R_{i+1}^{(t+1)}$ is the reward of the next action based on $S_{i+1}^{(t+1)}$. This formulation forces the critic to estimate the expected reward under the constraint that the next classification result on χ_{i+1} is sufficient and guarantees the compatibility of the adjacent prediction and the smoothness of the retrieved ground surface.

2.2.4. Network Architecture

The actor network aims to assign predictions to each point in the points-profile, which is a binary-segmentation task. We utilize the Minkowski U-net architecture and further integrate the proposed HCF module for increasing the receptive field and capturing a large context. The original Minkowski U-net [49] comprises four-level down-sampling blocks and corresponding up-sampling blocks with the same tensor stride. The HCF module is stacked to the bottom of the U-net and contains one strided sparse convolution and corresponding strided sparse transpose convolution, which between them is two residual convolution blocks, with each block containing two sparse convolution layers with a kernel size of $3 \times 3 \times 1$ and dilation factor of 2 and 3 for enlarging the receptive field.

Notably, rather than using regular sparse convolution, which has the same tensor stride in all three axes, we only use the $3 \times 3 \times 1$ convolution kernel in the HCF module, which based on the observation that the large-scale ALS point clouds usually requires a different degree of down-sampling on the z-axis and the x–y plane. At the bottom level of the network, the receptive field along the z-axis is already enough and further down-sampling is cumbersome in some way. These are the main differences from simply making the network deeper, we conduct an experiment and demonstrate that naively deepening the network may not produce satisfactory results. The network architecture is depicted in Figure 5.

2.3. Implementation Details

The training and testing sets are partitioned into patches by a sliding window with overlap, the window size is determined by the voxel size and input size of the network, and the overlap is a quarter of the window size. Only the prediction result without overlap is used as the final result during the testing phase for avoiding the edge effect. Random rotation around the z-axis is applied in the training stage for data augmentation. Hyper-parameters r and d which control the number of points-profiles in each profile-stack and the thickness of each points-profile are set to 1.0 m and 6, respectively, for training and testing. We only select Test Site II without outliers in OpenGF for comparison because the outliers have limited influences on the deep-learning-based GF method according to Qin et al. [6].

The whole experiment is conducted on Ubuntu 18.04 with Pytorch 1.7.1. Adam [68] is used for optimization with a batch size of 4. The actor and the critic network are updated in turn at each iteration. The size of the repay memory is set to 3200. It takes about 72 h for 60,000 iterations on an Intel 6700HQ CPU and an NVIDIA RTX3090 GPU.

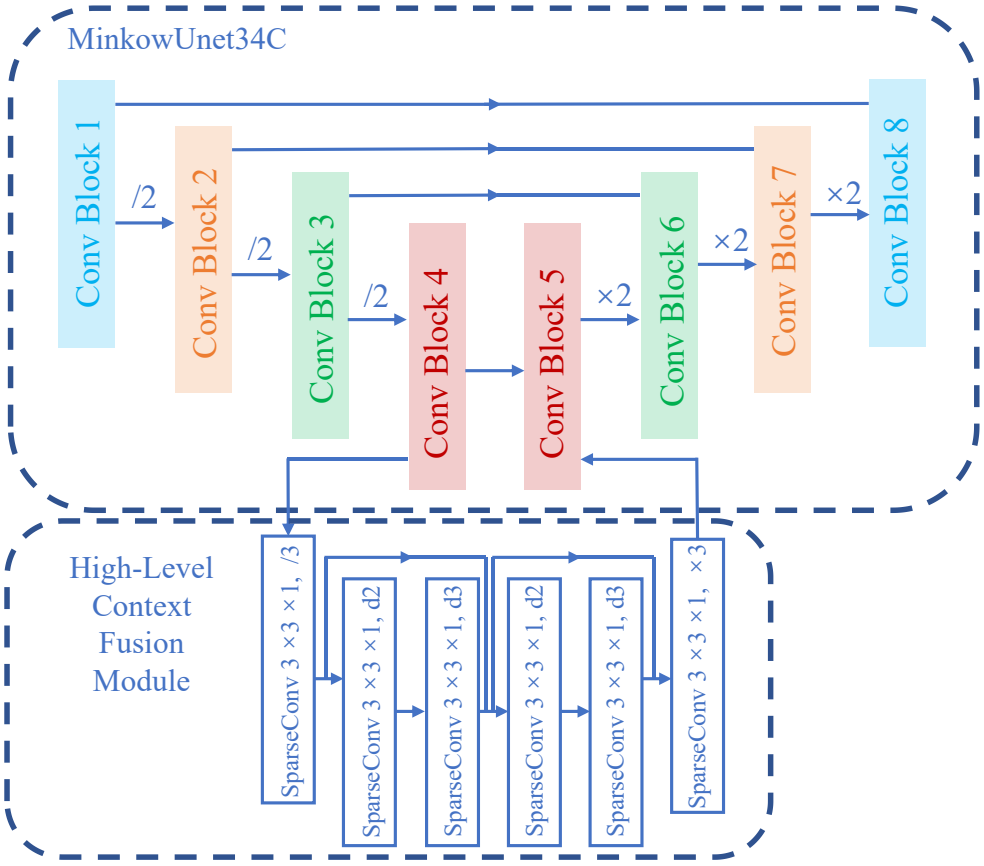

Figure 5. Network Architecture. Where /2 and ×2 denote down-sampling and up-sampling, respectively, with tensor stride size of 2, d2 and d3 denote dilation factor of 2 and 3, respectively, for sparse dilated convolution.

2.4. Evaluation Metrics

The Overall Accuracy (OA), Intersection over Union (IoU), Root Mean Square Error (RMSE), Matthews Correlation Coefficient (MCC), and Kappa Coefficient (KC) are adopted for evaluation [6]. Let TP_1/FP_1 be the number of non-ground points with correct/incorrect classification, TP_2 and FP_2 are similarly defined for ground points. The calculations are as follows:

$$OA = \frac{TP_1 + TP_2}{TP_1 + FP_1 + TP_2 + FP_2} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(E_i - R_i)^2}{N}} \tag{8}$$

$$IoU_1 = \frac{TP_1}{TP_1 + FP_1 + FP_2} \tag{9}$$

$$IoU_2 = \frac{TP_2}{TP_2 + FP_2 + FP_1} \tag{10}$$

$$MCC = \frac{TP_1 * TP_2 - FP_1 * FP_2}{\sqrt{(TP_1 + FP_1) * (TP_1 + FP_2) * (TP_2 + FP_1) * (TP_2 + FP_2)}} \tag{11}$$

$$P_e = \frac{(TP_1 + FP_2) * (TP_1 + FP_1) + (TP_2 + FP_1) * (TP_2 + FP_2)}{(TP_1 + FP_1 + TP_2 + FP_2)^2} \tag{12}$$

$$KC = \frac{OA - P_e}{1 - P_e} \tag{13}$$

IoU_1 and IoU_2 denote the intersection over the union of the non-ground class and ground class. N denotes the number of pixels while E_i and R_i denote the corresponding elevation value in the generated and ground truth DEM.

3. Experimental Results

In this section, we first make a comparison with the Minkowski sparse convolutional neural network in Section 3.1 to illustrate the conflict between the spatial resolution and contextual information and demonstrate the advantages of the SeqGP to acquire high resolution with large context. In Section 3.2, we make comparisons to the baseline methods of the OpenGF dataset, and further add the Dynamic graph CNN (DGCNN) [38] and SCF-Net [69] for comparison. Section 3.3 includes the ablation studies. Finally, we examine the generalization ability of the proposed methods in Section 3.4.

3.1. Comparisons with the Baseline Methods

We conduct an experiment based on MinkowUnet34C under different settings to demonstrate its efficiency and drawbacks when dealing with the GF problem. We set the spatial resolution (voxel-size) at 0.5 m, 1.0 m, and 1.5 m, and the corresponding spatial context of 128 m, 256 m, and 384 m. Thus, the maximum size of the input sparse tensor is fixed at 256^3.

As shown in Figure 6, the network with the highest spatial resolution (0.5 m) produces a fine result on Test site I; however, the large building roof, which spans more than 300 m, has not been removed properly due to the lack of contextual information. As contextual information increases, the large building roofs are correctly identified, but the incorrect classification close to the ground surface increases at the same time. In theory, we could set the high spatial resolution and large contextual information simultaneously (e.g., 0.5 m voxel-size with 384 m contexts), but the memory footprint may rise unacceptably, as shown in Figure 7. We set batch size 1 for a single forward pass and gain more than 20 Gb maximum GPU memory consumption on Test Site II. On Test Site I, 24 Gb GPU memory is insufficient to complete the whole prediction procedure when set at a spatial resolution (voxel-size) of 0.5 m and 0.75 m with 384 m contextual information. Due to the intrinsic characteristics of the sparse convolutional neural networks, GPU memory usage fluctuates with data sparsity. Thus, we only consider the maximum value.

The quantitative comparison is shown in Table 2, our method obtains the highest OA and lowest RMSE on both sites. The proposed method also obtains higher MCC and KC compared with the baselines, which indicates a better classification result. The performance is on par with the MinkowUnet with 0.5 m voxel size on Test Site I, which is mainly covered by vegetation and small buildings. In this situation, a small scale of contextual information (e.g., 128 m) is enough to determine non-ground objects. Notably, the proposed method surpasses the baseline methods on Test Site II significantly on OA, RMSE, MCC, and KC. Since an overly large context is required to remove large-scale objects (e.g., metropolitan buildings). The proposed method shows superiority in dealing with a hybrid scenario due to the ability to acquire high-resolution and large contextual information simultaneously under limited computational resources.

Table 2. Comparison with MinkowUnet-based segmentation methods under different settings.

Method	Test Site I						Test Site II					
	OA	***RMSE***	***IoU_1***	***IoU_2***	***MCC***	***KC***	***OA***	***RMSE***	***IoU_1***	***IoU_2***	***MCC***	***KC***
MinkowUnet (0.5)	96.45	0.27	93.84	92.28	92.82	92.81	92.31	3.26	84.85	86.49	85.38	84.62
MinkowUnet (1.0)	93.84	0.27	89.62	86.86	87.49	87.49	93.15	1.47	86.70	87.63	86.64	86.31
MinkowUnet (1.5)	90.33	0.34	84.76	79.09	80.59	80.13	91.74	0.55	84.11	85.32	83.87	83.49
Ours	96.52	0.23	94.12	92.15	93.05	92.90	95.20	0.32	90.90	90.78	90.40	90.40

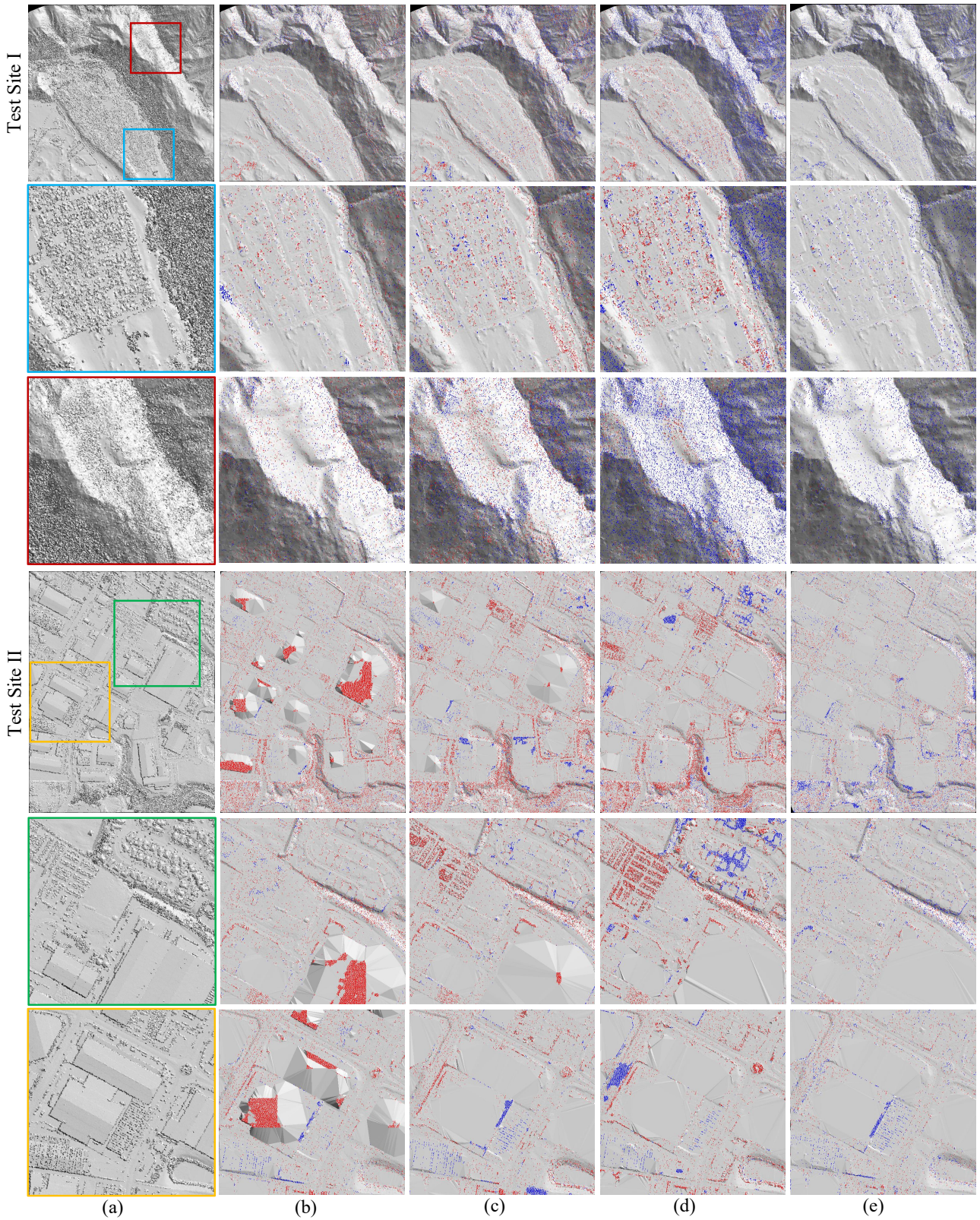

Figure 6. Visualization of MinkowUnet34C-based segmentation methods under different settings. (**a**) DSM, (**b**) 0.5 m voxel-size, (**c**) 1.0 m voxel-size, (**d**) 1.5 m voxel-size, (**e**) Ours (0.5 m voxel-size); Points in red and blue denote the misclassified non-ground and ground points, respectively.

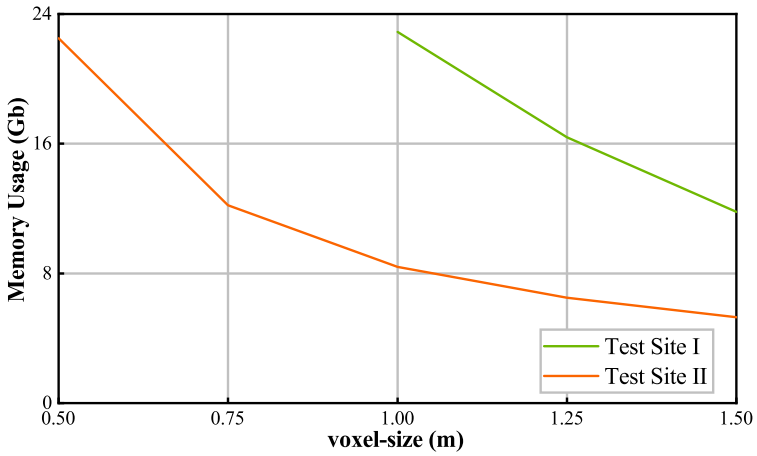

Figure 7. Maximum memory usage of MinkowUnet34C-based segmentation method on the test set, with a batch size of 1 and a coverage of 384 m for one forward pass.

3.2. Comparisons with State-of-the-Art Methods

We compare the deep-learning-based baseline of the OpenGF dataset, which include PointNet++ [36], KPConv [39] and RandLA-Net [40]. We further conduct an experiment on MinkowUnet34C [49] follow the configurations in [6], the voxel size for down-sampling is set to 1.0 m. The Dynamic graph CNN (DGCNN) [38] and SCF-Net [69] are further added for comparison. As shown in Figure 8, these methods produce the wrong classifications on the large building roofs in Test Site II except SCF-Net. The KPConv, PointNet++, and DGCNN fail to remove the large-scale man-made object due to the small sampling region (contextual information) with a 1.0 m grid size, while RandLA-Net and MinkowUnet produce relatively better results with fewer errors on large building roofs. The SCF-Net can properly remove the large building, but the performance on Test Site I are relatively lower than other baseline methods, the proposed method achieves lower RMSE than SCF-Net on the two test sites. Notably, the proposed method obtained the best result on Test Site II with large building roofs correctly removed. The quantitative results can be seen in Table 3. In Test Site II, the proposed method outperforms the others, while the RMSE is ahead of theirs significantly, demonstrating the advantages of the proposed method in dealing with large-scale man-made objects while keeping terrain details. In Test Site I, our method surpasses DGCNN, RandLA-Net, SCF-Net, and MinkowUnet34C on all the evaluation metrics but is a little bit lower than the best result produced by KPConv. The main reason is that KPConv obtains high spatial resolution with point-based representation, while sacrificing the contextual information, which is caught in the dilemma mentioned in Section 1, and thus produces obvious errors in Test Site II on building roofs, which deteriorates the final DEM results.

Table 3. Comparison with the baseline methods of OpenGF.

Method	Test Site I				Test Site II			
	OA	*RMSE*	IoU_1	IoU_2	*OA*	*RMSE*	IoU_1	IoU_2
PointNet++	97.58	0.25	95.75	94.68	87.38	4.89	75.19	79.63
DGCNN	96.34	0.41	93.78	91.81	93.86	3.59	88.16	88.68
KPConv	97.79	0.20	96.10	95.17	91.09	3.87	82.44	84.67
RandLA-Net	96.29	0.29	93.74	91.65	94.96	1.20	90.38	90.42
SCF-Net	95.92	0.83	92.97	91.14	95.21	0.95	90.66	91.04
MinkowUnet	93.84	0.27	89.62	86.86	93.15	1.47	86.70	87.63
Ours	96.52	0.23	94.12	92.15	95.20	0.32	90.90	90.78

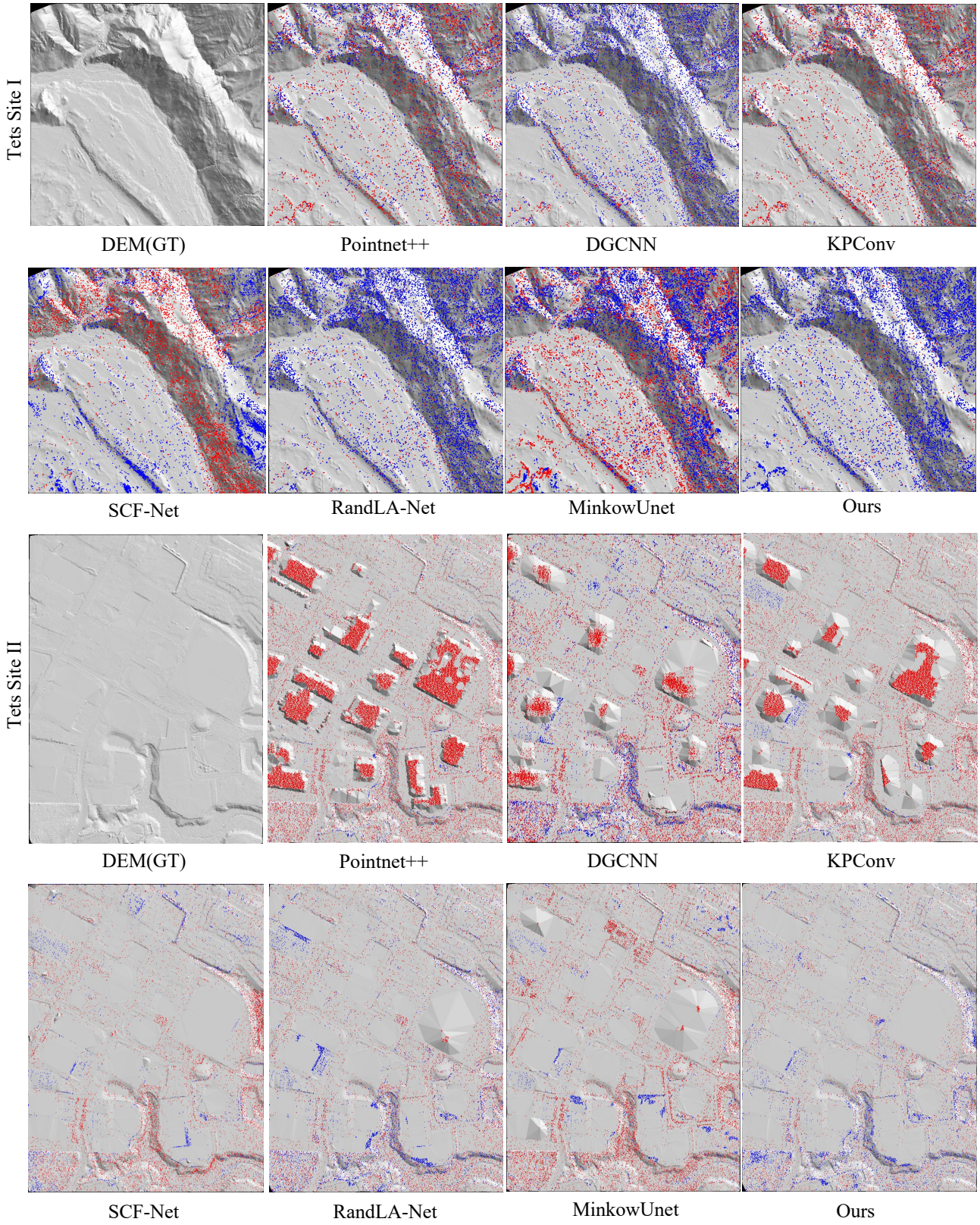

Figure 8. Comparison with the baseline methods of the dataset; Points in red/blue denote the misclassified non-ground/ground points, respectively; Results of these baseline methods are provided along with the OpenGF project [6].

3.3. Ablation Study

3.3.1. Module Effectiveness

We further study how the HCF module and the SeqGP strategy may affect the filtering result. We integrate the MinkowUnet34C segmentation method described in Section 3.1 with the proposed HCF module and SegGP strategy and evaluate the performance of the OpenGF dataset. As shown in Table 4, the iterative SeqGP strategy brings a 2.11 and 1.13 percentage improvement to the OA on Test Site I and Test Site II, which gives credit for the memory efficiency of SeqGP, making it possible to obtain a high spatial resolution with large contextual information, the detailed comparison with MinkowUnet based segmentation methods are given in Section 3.1. The HCF module further increases the OA by 0.57 and 0.92 percent, which is due to the HCF module bringing a larger receptive field at the bottom of the network and leading to better awareness of contextual information.

Table 4. Ablation studies of the proposed algorithm.

Network	OA(%)	
	Test Site I	Test Site II
MinkowUnet34C	93.84	93.15
MinkowUnet34C + SeqGP	95.95	94.28
MinkowUnet34C + HCF + SeqGP	96.52	95.20

3.3.2. Further Comparison of Different Network Architectures

We integrate the proposed SeqGP with different network architectures and demonstrate the effectiveness of the HCF module. First, we extend the MinkowUnet34C, which has four levels of residual convolution blocks to a deeper analog, which has one more down-sampling operation and comprises five levels of residual convolution blocks. The feature dimension in the fifth level is set to 512, which is doubled as it is in the fourth level. This is the simplest way to make the network deeper and obtain a larger receptive field. Second, we replace the fifth level of residual convolution blocks with the proposed HCF module. The performances of the above networks are evaluated and the original MinkowUnet34C stands as a baseline.

As shown in Figure 9, the proposed HCF module facilitates the removal of the large building roof significantly. The explanations are that, when dealing with a large-scale ALS point cloud, the points span at a larger range on the x–y plane, rather than on the z-axis, which means that the effective down-sampling ratio is different for the z-axis and x/y-axis. Especially at the bottom level of the network, the receptive field along the z-axis is sufficient. In this situation, down-sampling is redundant and is likely to result in misclassifications on flat terrains and large building roofs for the similarity of these two classes. Thus, the proposed HCF module, which only conducts the down-sampling operation on the x-axis and y-axis by a $3 \times 3 \times 1$ convolution kernel with dilation, produces better results. Notably, the number of parameters in the model with the HCF module is 237 M, which is almost half of the five-level MinkowUnet34C (408 M).

3.3.3. Hyper-Parameters

Hyper-parameters r and d control the number of points-profiles in each profile-stack and the thickness of each points-profile, respectively. The width of the profile-stack is calculated as

$$W_p = (2r + 1) * d \tag{14}$$

We fix d and W_p to evaluate the influence of different r as shown in Table 5. On the one hand, when fixing d to 1.0 m, a different value of r changes the contextual information contained in the profile-stack, the best OA on Test Site I and Test Site II is achieved by $r = 4$ and $r = 8$, which indicates that the optimal range of the context might change according to different scenes. However, when setting r to 2, which leads to a relatively small context, the performance degrades on Test Site I and Test Site II, while the other

settings only cause small fluctuations on OA. This indicates that the contextual information needs to be sufficient along the slicing direction ($W_p \geq 9$ m). For a fair comparison, we set $r = 6$, which is between $r = 4$ (best for Test Site I) and $r = 8$ (best for Test Site II), in the other experiments. When the r is set to 6, the W_p is fixed to 13 m, which we believe may produce fine results in most cases.

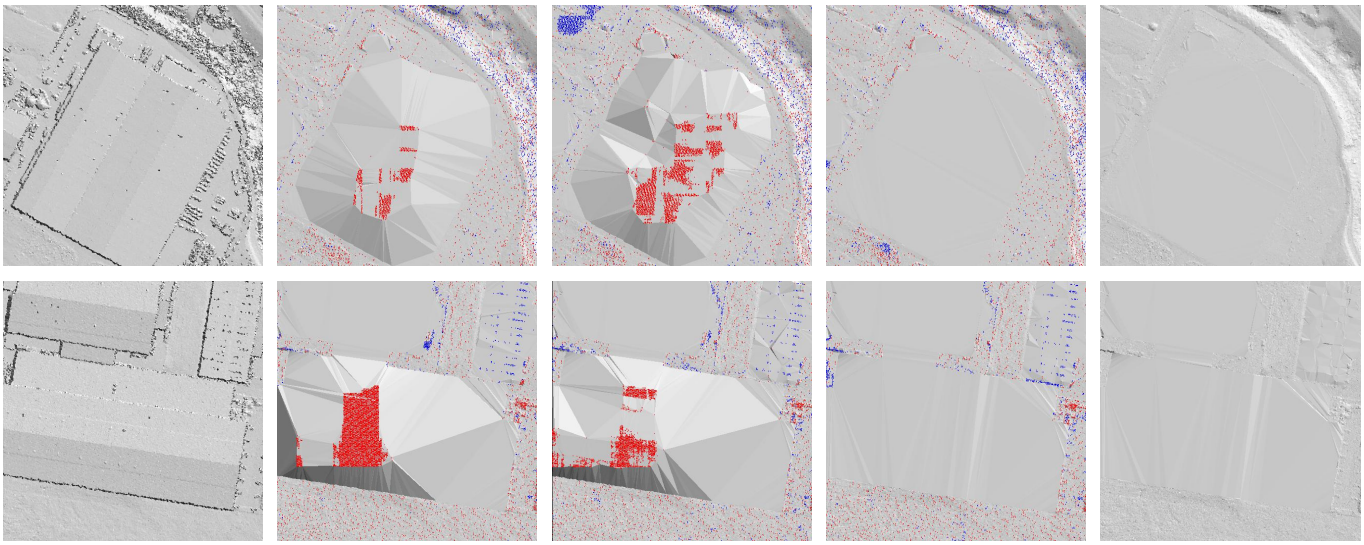

Figure 9. Some visualized results on large buildings in Test Site II of different network architectures. Points in red and blue denote the misclassified non-ground and ground points, respectively.

Table 5. Influence of number of points-profiles in profile-stack r with fixed width of the profile-stack W_p to 13 m.

Number of Points-Profiles	OA(%)			
	d = 1.0 m		W_p = 13.0 m	
	Test Site I	Test Site II	Test Site I	Test Site II
$r = 2$	94.89	88.76	96.77	95.75
$r = 4$	96.62	95.08	96.64	95.39
$r = 6$	96.52	95.20	96.52	95.20
$r = 8$	96.52	95.45	96.67	95.38
$r = 10$	96.53	95.29	96.64	95.39

On the other hand, when fixing W_p to 13 m, a different value of r may affect the thickness of the points-profile, and may lead to a different number of iterations for the sequential ground prediction process. The thicker the points-profile is, the fewer iterations would be needed to retrieve all ground points because more points are classified in one prediction step. The best OA on Test Site I and Test Site II are both achieved by $r = 2$, which indicates that thicker points-profile even led to better performance. No significant reduction has emerged in OA with different settings. Therefore, we set $r = 6$, which produces relatively lower OA on both test sites than other settings, in the other experiments to demonstrate the effectiveness of SeqGP.

The experiments demonstrate the robustness of the proposed method. With sufficient contextual information ($W_p \geq 9$ m), the larger d of points-profile thickness (smaller r) will improve the efficiency without performance losses.

3.4. Generalization Ability

We evaluate the proposed method on six testing areas in [34]. These areas are collected from southern China while the OpenGF is collected from four different countries. Thus, the test set shares a great domain gap with OpenGF on both terrain situations and the point density.

The comparison is made with two classic GF methods to demonstrate the generalizability of the proposed SeqGP, including PMF [9] and CSF [14]. The implementation of PMF in PDAL (https://pdal.io, accessed on 28 June 2021) and that of CSF in CloudCompare (https://www.cloudcompare.org, accessed on 30 March 2022) are adopted, while several combinations of parameters are evaluated to obtain the best results. For PMF, we tune the maximum window size (10, 20) with slope parameters (0.1, 0.5, 1.0) and cell size (0.5, 1.0, 2.0). For CSF, we set the scene choice of Steep slope or Relief with the corresponding cloth resolution (0.5, 1.0, 2.0). The best results on OA are selected for comparison. As shown in Table 6, the proposed method achieves the best performance, which demonstrates its generalizability to some extent. Some visualized results are shown in Figure 10.

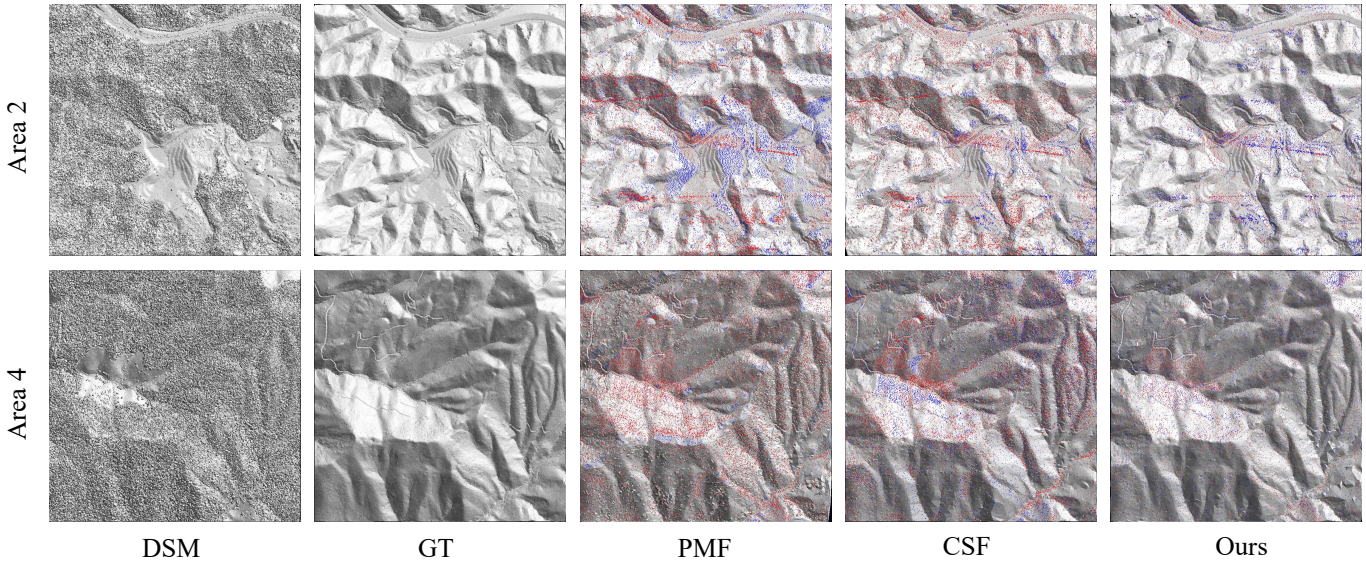

Figure 10. Some visualized results on Southern China dataset. Points in red and blue denote the misclassified non-ground and ground points, respectively. GT denotes the Ground Truth while PMF and CSF denote the Progressive Morphological Filter and the Cloth Simulation Filter, respectively.

Table 6. Comparison with rule-based methods on the Southern China dataset.

Methods	OA(%)					
	Area1	**Area2**	**Area3**	**Area4**	**Area5**	**Area6**
PMF	88.67	90.06	85.77	91.66	81.38	70.45
CSF	89.23	87.84	82.00	90.56	81.94	73.36
Ours	92.9	93.78	85.68	95.49	87.84	77.55

4. Discussion

4.1. Memory Efficiency

In order to deal with large-scale buildings while keeping high spatial resolution for details of the bare terrain, a large spatial context with a great number of points needs to be fed into the deep network in a forward pass. As shown in Section 3.1, for the sparse-convolution-based baseline, a compromised choice of voxel resolution and spatial context lacks universality. A high voxel resolution (e.g., 0.5 m) leads to a better performance in mountain areas while sacrificing spatial context, thus performing poorly in areas containing large buildings. Meanwhile, a large spatial context (e.g., 384 m) leads to coarse voxels under

limited GPU memory as shown in Figure 7, thus resulting in more misclassifications in the mountain areas. The best performance achieved by the baseline method is based on the settings of 1.0 m voxel size with 256 m spatial context, which is a compromised choice and demonstrates the conflict between the spatial resolution and the contextual information. For the point-based methods, the KPConv [39] and RandLA-Net [40] perform spatial grid sub-sampling to reduce the data volume. Similar to voxel-based methods, a large grid size of sub-sampling may lose geometric information, and the grid size is set to 1.0 m for the baseline methods of the OpenGF dataset [6]. Comparatively, the proposed GF pipeline takes points-profile as feature extraction inputs and reduces the memory consumption significantly, which makes it possible to process 384 m spatial context with 0.5 m voxel size under limited GPU memory.

4.2. Slicing Direction

Theoretically, the points-profiles can be sliced along any horizontal direction, but in practice, the large-scale ALS data is partitioned into square patches along x and y directions in most cases. Thus, we only consider the x and y directions for points-profile generation in this study. Throughout the whole experiment, the sequential ground prediction is conducted in both x and y directions and the final result is obtained by equally weighting the prediction probability of two directions. The influence of the slicing direction is shown in Table 7. The different slicing direction causes small fluctuations in OA, which indicates that the proposed method is not sensitive to the slicing direction and demonstrates the robustness of SeqGP to some extent.

A more reasonable way to slice the points-profile is by considering the scan-line information. However, the scanning direction is not always available in practice and multiple scanning directions are commonly present. The scan-line information is not provided in the datasets used in our study, thus we only use the x and y axis for points-profile generation.

Table 7. Influence of slicing directions.

Slicing Direction	OA(%)	
	Test Site I	Test Site II
x	95.80	94.33
y	95.86	94.50
$x + y$	96.52	95.20

4.3. Filtering Performance

Compared with the sparse-convolution-based baseline, the proposed GF pipeline achieves the best performance on the evaluation metrics of OA and RMSE. The OA and RMSE of the proposed method surpass that of the baseline method with 0.5 m voxel size on Test Site I slightly, which indicate that though the original point clouds are partitioned into points-profile, the correlation of adjacent profiles is recovered by the SegGP strategy; thus, the classification performance under high spatial resolution is preserved. Meanwhile, the proposed method outperforms the baselines significantly on Test Site II, which is contributed to the ability of the proposed method for handling large spatial contexts with high voxel resolution.

Compared with the baselines of OpenGF, the proposed SeqGP achieves state-of-the-art performance on Test Site II, the RMSE of the proposed method surpasses that of KPConv [39] and RandLA-Net [40] by 3.55% and 0.88%, respectively. Since the RMSE metric measures the difference between the final DEM produced by GF and the ground truth, it can evaluate the filtering performance more fairly than the OA metric. The proposed method keeps the RMSE under 1% on Test Site II for the first time, which demonstrates the advantage of the proposed method when dealing with large-scale ALS data. On Test Site I, the RMSE of the proposed method (0.23%) ranks second, slightly below that of KPConv by

0.03%, which indicates that the quality of the final DEM produced by the two methods is about the same.

4.4. Network Architecture

A large spatial context with high voxel resolution leads to a large spatial size of the input sparse voxels, on which a large receptive field is required for the neural network to obtain global information. One simple approach is to extend the existing network structure and build a deeper network. However, as shown in Section 3.3.2, directly extending the MinkowUnet34C to a deeper analog may not obtain a satisfactory result. When it comes to large-scale ALS point clouds, the spatial range of the horizontal direction mostly spans larger than that of the vertical direction, which means that the effective down-sampling ratio is different for the horizontal and vertical directions. In the meantime, to recognize large buildings, the size of the required receptive field is larger in the horizontal direction than in the vertical one. The proposed HCF module is motivated by the above observations, the down-sampling operations in the HCF module are only performed horizontally, which enlarges the receptive field along the horizontal direction while avoiding the information losses in the vertical direction. The HCF module is stacked to the bottom of the network because the down-sampling along both horizontal and vertical directions is necessary for the early stage, the HCF module is only supposed to acquire a large receptive field along the horizontal direction at the deepest level of the network. As can be seen from Figure 9, the large buildings are properly recognized with the help of the HCF module.

4.5. Performances in Different Land Use Classes

We further analyze the performances of SeqGP in various land use classes. Since the land cover annotations are absent in OpenGF, we manually collect three classes from the test sites for discussion, including forests (steep areas), grasslands, and croplands. There are ten samples in each class of forest and grassland, while the cropland class contains six. The quantitative results, which are calculated by averaging across samples in each class, can be seen in Table 8. On the whole, the SeqGP performs well in most situations. In croplands, the SeqGP achieved an MCC of 91.81% and KC of 91.71%, which indicates the classification result is good enough in this class. The main reason is that the croplands are relatively more structured with fewer complex terrain situations than the other land use classes, and thus easier for the network to distinguish. In grasslands, the IoU_1 is lower than the other classes. Since these areas are mainly covered by sparse and low vegetation, the amount of ground points is much larger than that of non-ground points in these areas. Therefore, the wrong classifications in non-ground points may cause obvious fluctuations of IoU_1. Some near-ground points of low vegetation are easy to be misclassified since the voxel size is 0.5 m in our experiments. Further increasing the spatial resolution may alleviate this problem. In steep forested areas, the GF performances degrade to some extent. Because these areas are mainly covered by dense forests, the ground points are relatively sparse. In this situation, the SeqGP are more sensitive to the direction of points-profile, using multiple slicing directions and assembling prediction strategy may improve the performance.

Table 8. Performances in different land use classes.

Land Cover	*OA* (%)	*RMSE*	IoU_1(%)	IoU_2(%)	*MCC* (%)	*KC* (%)
Forests	96.26	0.19	95.13	79.03	86.21	85.64
Grasslands	95.08	0.05	85.11	92.32	88.29	87.58
Croplands	96.21	0.02	90.38	93.78	91.81	91.71

The per-sample evaluations are illustrated in Figure 11, in which we can observe more variance in each metric in the steep forested areas than in the other two classes. It demonstrates that the SeqGF is more robust in grasslands and croplands. Meanwhile, the performance in steep forested areas is relatively weak. However, please note that

the highest RMSE among the ten forest samples is around 0.2, which indicates a fine performance according to the result of Tables 2 and 3.

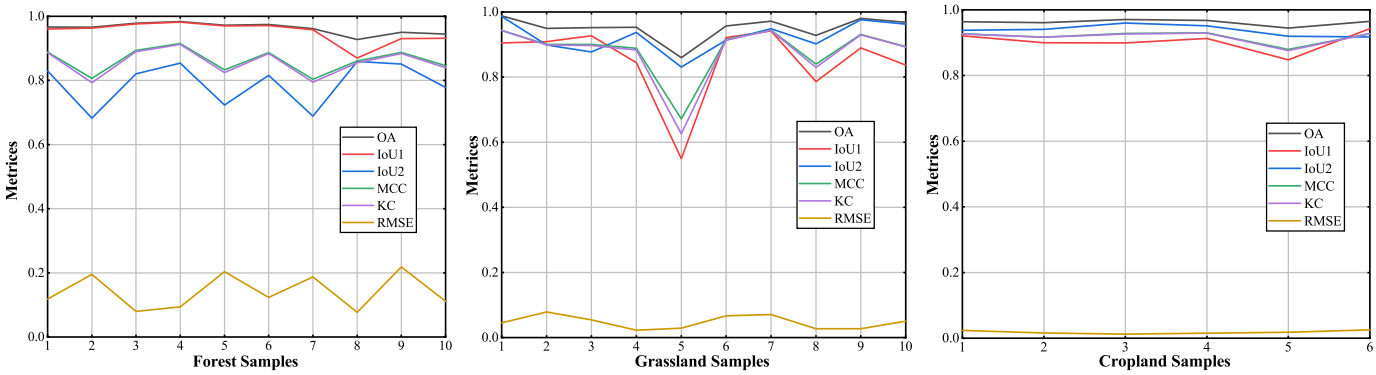

Figure 11. Per-sample performances in different land use classes.

4.6. Limitations and Future Work

Overall, the proposed SeqGP achieved competitive performance of ground filtering on large-scale ALS point clouds. The extensive experiments demonstrate the universality of the SeqGP when dealing with various scenes. However, there are still some limitations of our study that can be improved in future research. First, the sparse voxel representation has a unique advantage in dealing with large data volumes, but the voxelization process may lose geometric details. In the meantime, the point-based methods can preserve the original geometric information but have more limited spatial coverage. Incorporating the advantages of the voxel and point representation is a promising direction for future research on large-scale ALS point cloud processing. Second, as mentioned in Section 4.2, the scan-line direction may be valuable information for points-profile partition, we plan to examine this issue in more detail in future research.

5. Conclusions

A deep-learning-based GF framework dedicated to large-scale ALS point clouds is investigated in this paper. We propose an iterative SeqGP strategy that utilizes DRL algorithms to retrieve ground points based on the points-profile data organization. The proposed method achieves state-of-the-art performance on the challenging test set of OpenGF and shows good generalization ability on the Southern China dataset.

The proposed framework brings a novel solution to the large-scale GF problem. Under limited computational resources, the proposed SeqGP alleviates the conflict between keeping high spatial resolution and acquiring a large semantic context. Incorporating with the HCF module, the proposed framework can remove large-scale man-made objects properly while preserving sharp details of the bare terrain.

Author Contributions: H.D. conceptualized the framework, conducted the experiments, and also wrote the manuscript. X.H. supervised and revised this manuscript. Z.S. and N.Q. assisted in the experiments and also revised the manuscript. J.Z. revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Special Fund of Hubei Luojia Laboratory under grant 220100028, and the National Natural Science Foundation of China under grant 42001400.

Data Availability Statement: The data presented in this study are available on request from the first author.

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

References

1. McCarley, T.R.; Hudak, A.T.; Sparks, A.M.; Vaillant, N.M.; Meddens, A.J.; Trader, L.; Mauro, F.; Kreitler, J.; Boschetti, L. Estimating wildfire fuel consumption with multitemporal airborne laser scanning data and demonstrating linkage with MODIS-derived fire radiative energy. *Remote Sens. Environ.* **2020**, *251*, 112114. [CrossRef]
2. Stereńczak, K.; Kraszewski, B.; Mielcarek, M.; Piasecka, Ż.; Lisiewicz, M.; Heurich, M. Mapping individual trees with airborne laser scanning data in an European lowland forest using a self-calibration algorithm. *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *93*, 102191.
3. Doneus, M.; Mandlburger, G.; Doneus, N. Archaeological ground point filtering of airborne laser scan derived point-clouds in a difficult mediterranean environment. *J. Comput. Appl. Archaeol.* **2020**, *3*, 92–108.
4. Mezaal, M.R.; Pradhan, B.; Rizeei, H.M. Improving landslide detection from airborne laser scanning data using optimized Dempster–Shafer. *Remote Sens.* **2018**, *10*, 1029.
5. Nie, S.; Wang, C.; Dong, P.; Xi, X.; Luo, S.; Qin, H. A revised progressive TIN densification for filtering airborne LiDAR data. *Measurement* **2017**, *104*, 70–77. [CrossRef]
6. Qin, N.; Tan, W.; Ma, L.; Zhang, D.; Li, J. OpenGF: An Ultra-Large-Scale Ground Filtering Dataset Built Upon Open ALS Point Clouds Around the World. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 1082–1091.
7. Vosselman, G. Slope based filtering of laser altimetry data. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 935–942.
8. Wang, C.; Tseng, Y. DEM gemeration from airborne lidar data by an adaptive dualdirectional slope filter. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 628–632.
9. Zhang, K.; Chen, S.C.; Whitman, D.; Shyu, M.L.; Yan, J.; Zhang, C. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 872–882.
10. Chen, Q.; Gong, P.; Baldocchi, D.; Xie, G. Filtering airborne laser scanning data with morphological methods. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 175–185.
11. Axelsson, P. DEM generation from laser scanner data using adaptive TIN models. *Int. Arch. Photogramm. Remote Sens.* **2000**, *33*, 110–117.
12. Kraus, K.; Pfeifer, N. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* **1998**, *53*, 193–203.
13. Błaszczak-Bąk, W.; Janowski, A.; Kamiński, W.; Rapiński, J. Application of the Msplit method for filtering airborne laser scanning data-sets to estimate digital terrain models. *Int. J. Remote Sens.* **2015**, *36*, 2421–2437.
14. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501.
15. Pfeifer, N.; Mandlburger, G. LiDAR data filtering and DTM generation. In *Topographic Laser Ranging and Scanning*; CRC Press: Boca Raton, FL, USA, 2017; pp. 307–334.
16. Chen, Z.; Gao, B.; Devereux, B. State-of-the-art: DTM generation using airborne LIDAR data. *Sensors* **2017**, *17*, 150. [CrossRef] [PubMed]
17. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [PubMed]
18. Cutler, A.; Cutler, D.R.; Stevens, J.R. Random forests. In *Ensemble Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 157–175.
19. Lafferty, J.; McCallum, A.; Pereira, F.C. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*; University of Pennsylvania: Philadelphia, PA, USA, 2001.
20. Kang, Z.; Yang, J.; Zhong, R. A bayesian-network-based classification method integrating airborne lidar data with optical images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *10*, 1651–1661. [CrossRef]
21. Zhang, J.; Lin, X.; Ning, X. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sens.* **2013**, *5*, 3749–3775.
22. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Classification of urban LiDAR data using conditional random field and random forests. In Proceedings of the Joint Urban Remote Sensing Event 2013, Sao Paulo, Brazil, 21–23 April 2013; pp. 139–142.
23. Schmidt, A.; Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of full waveform lidar data in the Wadden Sea. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1614–1618. [CrossRef]
24. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165.
25. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364.
26. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
27. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1887–1893.
28. Hu, X.; Yuan, Y. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sens.* **2016**, *8*, 730. [CrossRef]

29. Rizaldy, A.; Persello, C.; Gevaert, C.; Oude Elberink, S.; Vosselman, G. Ground and multi-class classification of airborne laser scanner point clouds using fully convolutional networks. *Remote Sens.* **2018**, *10*, 1723. [CrossRef]
30. Yang, Z.; Jiang, W.; Xu, B.; Zhu, Q.; Jiang, S.; Huang, W. A convolutional neural network-based 3D semantic labeling method for ALS point clouds. *Remote Sens.* **2017**, *9*, 936. [CrossRef]
31. Wang, B.; Wang, H.; Song, D. A Filtering Method for LiDAR Point Cloud Based on Multi-Scale CNN with Attention Mechanism. *Remote Sens.* **2022**, *14*, 6170.
32. Jin, S.; Su, Y.; Zhao, X.; Hu, T.; Guo, Q. A point-based fully convolutional neural network for airborne lidar ground point filtering in forested environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3958–3974.
33. Li, B.; Lu, H.; Wang, H.; Qi, J.; Yang, G.; Pang, Y.; Dong, H.; Lian, Y. Terrain-Net: A Highly-Efficient, Parameter-Free, and Easy-to-Use Deep Neural Network for Ground Filtering of UAV LiDAR Data in Forested Environments. *Remote Sens.* **2022**, *14*, 5798.
34. Zhang, J.; Hu, X.; Dai, H.; Qu, S. DEM extraction from ALS point clouds in forest areas via graph convolution network. *Remote Sens.* **2020**, *12*, 178. [CrossRef]
35. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
36. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv* **2017**, arXiv:1706.02413.
37. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
38. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 1–12. [CrossRef]
39. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
40. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11108–11117.
41. Janssens-Coron, E.; Guilbert, E. Ground point filtering from airborne lidar point clouds using deep learning: A preliminary study. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 1559–1565. [CrossRef]
42. Fareed, N.; Flores, J.P.; Das, A.K. Analysis of UAS-LiDAR Ground Points Classification in Agricultural Fields Using Traditional Algorithms and PointCNN. *Remote Sens.* **2023**, *15*, 483.
43. Nurunnabi, A.; Teferle, F.; Li, J.; Lindenbergh, R.; Hunegnaw, A. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Info. Sci* **2021**, *24*, 1–8.
44. Nurunnabi, A.; Teferle, F.; Li, J.; Lindenbergh, R.; Parvaz, S. Investigation of Pointnet for Semantic Segmentation of Large-Scale Outdoor Point Clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 4. [CrossRef]
45. Yotsumata, T.; Sakamoto, M.; Satoh, T. Quality improvement for airborne lidar data filtering based on deep learning method. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 355–360.
46. Wang, P.S.; Liu, Y.; Guo, Y.X.; Sun, C.Y.; Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.* **2017**, *36*, 1–11.
47. Klokov, R.; Lempitsky, V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 863–872.
48. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3d semantic segmentation with submanifold sparse convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9224–9232.
49. Choy, C.; Gwak, J.; Savarese, S. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3075–3084.
50. Schmohl, S.; Sörgel, U. Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 77–84. [CrossRef]
51. Huang, S.; Usvyatsov, M.; Schindler, K. Indoor scene recognition in 3D. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 8041–8048.
52. Gwak, J.; Choy, C.; Savarese, S. Generative sparse detection networks for 3d single-shot object detection. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 297–313.
53. Xie, S.; Gu, J.; Guo, D.; Qi, C.R.; Guibas, L.; Litany, O. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 574–591.
54. Hu, W.; Zhao, H.; Jiang, L.; Jia, J.; Wong, T.T. Bidirectional Projection Network for Cross Dimension Scene Understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 14373–14382.
55. Guo, B.; Li, Q.; Huang, X.; Wang, C. An improved method for power-line reconstruction from point cloud data. *Remote Sens.* **2016**, *8*, 36. [CrossRef]

56. Fan, J.; Ma, L.; Sun, A.; Zou, Z. An approach for extracting curve profiles based on scanned point cloud. *Measurement* **2020**, *149*, 107023. [CrossRef]
57. Xu, X.; Yang, H.; Neumann, I. Time-efficient filtering method for three-dimensional point clouds data of tunnel structures. *Adv. Mech. Eng.* **2018**, *10*, 1687814018773159. [CrossRef]
58. Sithole, G.; Vosselman, G. Filtering of airborne laser scanner data based on segmented point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2005**, *36*, W19.
59. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [PubMed]
60. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292.
61. Liu, F.; Li, S.; Zhang, L.; Zhou, C.; Ye, R.; Wang, Y.; Lu, J. 3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5678–5687.
62. Liao, X.; Li, W.; Xu, Q.; Wang, X.; Jin, B.; Zhang, X.; Wang, Y.; Zhang, Y. Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9394–9402.
63. Sarmad, M.; Lee, H.J.; Kim, Y.M. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5898–5907.
64. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
65. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
66. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
67. Feng, W.; Zhuo, H.H.; Kambhampati, S. Extracting action sequences from texts based on deep reinforcement learning. *arXiv* **2018**, arXiv:1803.02632.
68. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
69. Fan, S.; Dong, Q.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.Y. SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 14504–14513.