



Probability propagation for faster and efficient point cloud segmentation using a neural network

Hogeon Seo^{a,b,1}, Sangjun Noh^{c,1}, Sungho Shin^{c,1}, Kyoobin Lee^{c,*}

^a Korea Atomic Energy Research Institute, 111, Daedeok-daero 989beon-gil, Yuseong-gu, Daejeon, 34057, Republic of Korea

^b University of Science & Technology, 217, Gajeong-ro, Yuseong-gu, Daejeon, 34113, Republic of Korea

^c Gwangju Institute of Science and Technology, 123, Cheomdangwagi-ro, Buk-gu, Gwangju, 61005, Republic of Korea

ARTICLE INFO

Article history:

Received 9 January 2023

Revised 18 March 2023

Accepted 22 April 2023

Available online 24 April 2023

Edited by: Jiwen Lu

Keywords:

Neural network

Point cloud segmentation

Probability propagation

Stochastic upsampling

Sampling method

ABSTRACT

Neural networks (NN) have shown promising performance in point cloud segmentation (PCS). However, the measured points are too numerous to be used as model input at once. It results in a long inference time and high computational cost due to iterative sampling and inference. This study proposes *Probability Propagation* (PP) as a stochastic upsampling method. PP propagates the predicted probability of a sampled part of a point cloud into the other unpredicted points by considering proximity. By replacing the iterative inference of NN with PP, large point clouds can be dealt with quickly and efficiently. We investigated the effectiveness of PP using the ShapeNet benchmark on various settings: sampling methods (random, farthest point, and Poisson disk sampling) with sampling ratios (5%, 10%, 20%, 39%, and 78%) for NN and the stochastic mapping conditions (uniform, linear, cosine, Gaussian, and exponential distributions) for PP. Using NN with PP achieved higher performance and faster inference speed than when using NN alone. For the farthest point sampling method of 5% sampling ratio, NN+PP improved the instance mIoU by 2.457% with 102 times faster speed compared to that when using NN alone. The result indicates that PP can significantly contribute to the improvement of performance and efficiency in PCS when used in edge AI systems.

© 2023 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

As the demand for mobile robots and autonomous vehicles grows, so has the need for LiDAR (light detection and ranging) used to scan and understand their surroundings [1]. High-resolution 3D laser scanners are now widely used for various applications on the ground and in the air, such as drones and unmanned aerial vehicles [2]. In understanding the environment through a point cloud acquired by the laser scanner, it is crucial to recognize the distance and shape of the object and to distinguish each part individually for the intelligent manipulation of autonomous systems and proper real-time interaction with the surrounding environment [3].

Point cloud segmentation (PCS) is the task of dividing the point cloud into subsets with the same parts by classifying each point in the point cloud [4–7]. Compared to 2D segmentation of images, 3D PCS is highly challenging when the point cloud is not struc-

tured and ordered data in situations where point clouds continue to accumulate over time [8], such as mobile mapping. It also has high redundancy and nonuniform point density [2]. Following the success of deep learning in the image domain, many algorithms have been proposed for the PCS. For the early approaches, voxelized point clouds were segmented using 3D CNN [9,10]. However, they require substantial computational costs to represent the volumes. The following research [11] proposed the fully connected layer to input the point cloud directly into a neural network (NN). Though these methods have improved the PCS performances using the simple layers, they failed to capture local patterns within complex scenes. Recent studies found that utilizing feature aggregation modules, which extract and aggregate features by progressively enlarging receptive fields, helped preserve geometric details [12–14].

The introduced methods have improved the PCS performance; however, there are still remaining issues, in terms of field applicability, for edge AI systems with low-cost computing, such as mobile robots, autonomous vehicles, and self-driving drones. As the resolution of LiDAR increases, the number of scanned points becomes too large to be inferred at once by NNs [15]. The amount of memory available limits the model performance [16]. Inferring a

* Corresponding author.

E-mail addresses: hogeony@kaeri.re.kr (H. Seo), sangjun7@gm.gist.ac.kr (S. Noh), hogili89@gm.gist.ac.kr (S. Shin), kyoobinlee@gist.ac.kr (K. Lee).

¹ These authors contributed equally to this work.

large number of points with limited memory requires sampling some of the point clouds and inferring multiple times up to the entire point prediction [14], where a trade-off between memory and inference time exists.

This study proposes *Probability Propagation* (PP) as a stochastic mapping method. PP replaces expensive NN inferences with the propagation of the inferred probability distribution to save inference time along with reduced memory usage. We investigated the effectiveness and efficiency of PP by comparing the PCS performance of NN alone to that of NN with PP for various experiment settings: sampling methods for NN and weight function for PP, called stochastic mapping conditions. PointNet++ [12] was used as NN, and ShapeNet part segmentation dataset [17] was used as the dataset.

This study provides a simple solution for maintaining the superior segmentation performance of NN even with significantly reduced inference time and memory usage. Furthermore, in respect of the maintainability of devices, the more the graphic processor unit (GPU) is used for the segmentation of all the points in the point cloud, the less the battery life of the systems lasts [18]. Therefore, it is necessary to develop a method to perform the PCS for all points successfully while minimizing GPU usage. PP can contribute to reducing the computational cost of PCS required for the embedded AI systems by minimizing data transfer between edge devices and a high-performance GPU server and minimizing the number of inferences via GPUs.

2. Related works

2.1. Deep learning based point cloud segmentation

Segmentation of voxelized point clouds was an early approach for PCS based on deep learning. Maturana et al. proposed the VoxNet [9] which applied 3D CNN to detect target objects from voxelized point clouds. The irregularity and unstructured problems of a point cloud can be solved by voxelization. Tchapmi et al. presented the end-to-end PCS framework called SEGCloud [10]. Coarse segmentation results were estimated by using the voxelized point clouds. After the coarse segmentation, detailed parts were segmented via trilinear interpolation and refined via fully connected random fields. However, they have the disadvantage of containing a lot of unnecessary information because voxel structures store occupied spaces as well as free spaces, which requires huge memory usage [11].

Qi et al. presented the pioneering work, PointNet [11], which consists of max pooling and fully connected layers that consume a point cloud as input. Before the fully connected layers, they applied the simple symmetry function to the point sets to make a model invariant to permutation. Compared with the previous approaches [9,10], PointNet has advantages in its simplicity along with superior performance. However, it failed to capture local patterns within the complex scenes. Its follow-up study [12] applied hierarchical grouping and sampling approaches to progressively extract features of larger regions along the hierarchy. Similarly, Zhao et al. utilized self-attention maps to preserve the relationship among the points [13].

To expand the applicability of PCS algorithms even in limited resource environments, Hu et al. proposed RandLANet that merged the random sampling with local feature aggregation module [14] in an efficient manner. In order to minimize the possibility of missing key points during random sampling and preserve the geometric details, a local feature accumulation module was used to extract the relations among the sampled points. This model contributes to reducing the inference time and memory consumption required for the inferences using NN. However, the proposed approach has a limited attentive field, which may not be effective for capturing

contextual information for points that are far away from each other or are collected in detail and densely. Additionally, while RandLANet is focused on fast inference for down-sampled points, it still requires an upsampling method to segment the entire point cloud. As a result, we aim to develop a fast and accurate upsampling method that considers contextual information for all points.

3. Proposed methodology

This study investigates the performance and efficiency of PCS based on the proposed methodology using NN with PP (NN+PP). The overall procedure of NN+PP is depicted in Fig. 1. Instance mIoU are evaluated in terms of performance. Relative inference time and average iteration count are compared in terms of efficiency. We first state the problem definition that segments all the points in a point cloud using NN with a sampling method of a specific number of points. Next, we introduce our proposed approach with NN+PP, which significantly reduces the inference time required for all the points in the point cloud and even improves PCS performance. Multiple inferences via NN are replaced by PP in the proposed method, in which the prediction probability distributions of the points inferred by NN with the initial sampling are stochastically spread to the other unlabeled points to reduce computational costs.

3.1. Problem statement

For PCS using NN alone, points in a point cloud must be divided into the sampled points whose number fits the model input size. The points in the point cloud can be defined as

$$\mathbf{X} = \mathbf{X}^S \cup \mathbf{X}^U, \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{N \times 3}$ are all the points in the point cloud; $\mathbf{X}^S \in \mathbb{R}^{n \times 3}$ and $\mathbf{X}^U \in \mathbb{R}^{(N-n) \times 3}$ are the sampled and unsampled points, respectively. N is the total number of points in the point cloud and n is the number of sampled points.

For the sampled points, NN predicts their labels as a form of probability $p(y_i = C | \mathbf{x}_i; \mathbf{x}_i \in \mathbf{X})$, where y_i is the predicted label of the i th point, C is the ground truth label, and $\mathbf{x}_i$ is the x, y, z coordinates of the i th point. The probability distributions (p_{NN}) from NN represent the probability that a subset of sampled points $\mathbf{X}^S$ corresponds to their classes as follows:

$$p_{NN}(\mathbf{y}^S | \mathbf{X}^S) = \mathcal{H}(\mathbf{X}^S), \quad (2)$$

where $\mathcal{H}: \mathbf{X}^S \rightarrow \mathbf{y}^S$ is the model for the PCS and $\mathbf{y}^S = \{y_1^S, y_2^S, \dots, y_n^S\}$ are the predicted labels for the sampled points. If n is smaller than N , the model is required to be used as many times as the quotient of N divided by n for the segmentation of all points in the point cloud until $\mathbf{X}^U$ has no element; the PCS by iterative inference using NN is described in Algorithm 1.

However, the multiple inferences of the model are computationally expensive and time-consuming, especially when $N \gg n$. Instead of repetitive inferences, we propose a novel method to propagate the probability distributions of the initial sampled points to the neighboring unsampled points. The objective of our proposed approach is the estimation of a probability distribution that the i th unsampled point $\mathbf{x}_i^U$ is classified into their classes when the probability distributions for sampled points are given as follows:

$$p(y_i^U | \mathbf{x}_i^U; \mathbf{x}_i^U \in \mathbf{X}^U, p_{NN}(\mathbf{y}^S | \mathbf{X}^S)). \quad (3)$$

3.2. Probability propagation (PP)

We propose the PP algorithm to propagate the probabilistic distributions of the sampled points to the unsampled points, stochastically weighted by the normalized distance between them. PP approaches involve segmentation, accumulation, and normalization.

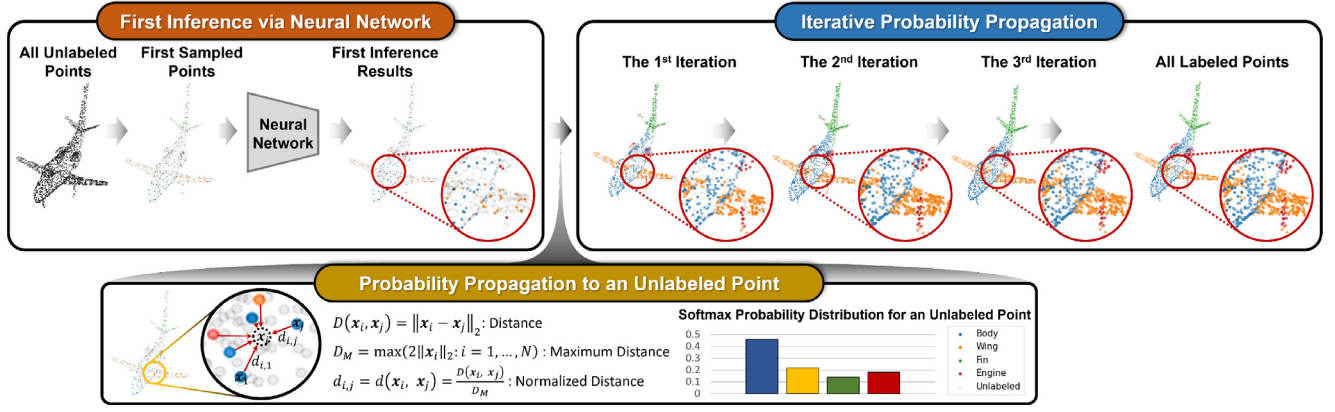


Fig. 1. The overall pipeline of probability propagation. The segmentation of all the points is conducted by single inference via a NN for its initial sampled points and then iterative probability propagation for the remaining points.

Algorithm 1 PCS by iterative inference using NN with sampling for all the points (NN).

```

1: Input: the coordinates  $(x, y, z)$  of a point  $\mathbf{x}$ , all the points of a
   point cloud  $\mathbf{X}$ , the unlabeled points  $\mathbf{X}^U$ , the number of points
   for the model input  $N^S$ , and the probability distributions  $p_{NN}$ 
2: Output: the predicted results  $\mathbf{y}$  for all the points  $\mathbf{X}$ 
3:  $N^U \leftarrow \text{length}(\mathbf{X}^U)$ 
4: while  $N^U > 0$  do
5:    $\mathbf{X}^S \leftarrow$  the sampled  $N^S$  points from  $\mathbf{X}^U$ 
6:    $\mathbf{X}^U \leftarrow \{\mathbf{x} \notin \mathbf{X}^S\}$ 
7:    $\mathbf{y}^S \leftarrow p_{NN}(\mathbf{y}^S | \mathbf{X}^S)$  as in Eq. (2)
8:    $\mathbf{y} \leftarrow \mathbf{y} \cup \mathbf{y}^S$ 
9:    $N^U \leftarrow \text{length}(\mathbf{X}^U)$ 
10:  if  $N^U < N^S$  then
11:     $\mathbf{X}^A \leftarrow (N^S - N^U)$  points from  $\{\mathbf{x} \in \mathbf{X}^S\}$ 
12:     $\mathbf{X}^U \leftarrow \mathbf{X}^U \cup \mathbf{X}^A$ 
13:     $N^U \leftarrow N^S$ 
14:  end if
15: end while

```

First, the segmentation model is utilized for the prediction of the initial sampled points. Then, the probability distribution of each unsampled point is estimated by accumulating the probability distributions of neighboring sampled points and normalizing the accumulated probability distribution.

For the segmentation, the probability distribution of the i th unlabeled point from the j th labeled point is obtained by the following equation:

$$p(y_i^U | \mathbf{x}_i^U; \mathbf{x}_i^U \in \mathbf{X}^U, p(y_j^S | \mathbf{x}_j^S; \mathbf{x}_j^S \in \mathbf{X}^S)) = S(d(\mathbf{x}_i^U, \mathbf{x}_j^S) | d_E) * p(y_j^S | \mathbf{x}_j^S), \quad (4)$$

where $S(\cdot)$ is the stochastic weight function. $d(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{D_M}$ is the normalized L_2 distance measure between the input points where D_M is the maximum distance obtained as twice the largest radius that is the distance between the centroid of the point cloud and its furthest point. Using D_M , the distance between the points was normalized to lie between 0 to 1. It may be unreasonable to use the maximum reference distance (D_M) as defined in this study when outliers exist very far away. Therefore, it is necessary to preprocess for handling outliers. It is also possible to normalize using a criterion other than necessarily the greatest distance considering the density and maximum distance of points. $S(d(\mathbf{x}_i^U, \mathbf{x}_j^S) | d_E)$ is the stochastic weight function that inputs the normalized distance between two points and then outputs the weight factor for the accumulation. d_E is the effective distance ratio for reflecting

neighboring sampled points only; if a normalized distance between the sampled and unsampled points is larger than the effective distance, the weight is set to 0 to exclude the influence of distant points. If PP does not occur due to insufficient d_E , d_E is temporarily increased by 1.5 at the next PP until PP occurs.

The probability distributions for an unsampled point obtained from sampled points are summed as follows:

$$A(\mathbf{x}_i^U, p(\mathbf{y}^S | \mathbf{X}^S)) = \sum_{j=1}^{N^S} (S(d(\mathbf{x}_i^U, \mathbf{x}_j^S) | d_E) * p(y_j^S | \mathbf{x}_j^S)), \quad (5)$$

where N^S is the number of labeled points.

Finally, the accumulated results are normalized for each iteration of PP by applying the softmax function to the accumulated probability distribution as follows:

$$p_{PP}(y_i^U | \mathbf{x}_i^U; \mathbf{x}_i^U \in \mathbf{X}^U, p(\mathbf{y}^S | \mathbf{X}^S)) = \begin{cases} 0 & \text{if } \|A(\mathbf{x}_i^U, p(\mathbf{y}^S | \mathbf{X}^S))\|_1 = 0 \\ \sigma(A(\mathbf{x}_i^U, p(\mathbf{y}^S | \mathbf{X}^S))) & \text{otherwise} \end{cases} \quad (6)$$

where $\sigma(\cdot)$ is the softmax function to normalize the probability distribution. If $p_{PP}(y_i^U | \mathbf{x}_i^U; \mathbf{x}_i^U \in \mathbf{X}^U, p(\mathbf{y}^S | \mathbf{X}^S)) = 0$, the probability distributions for a unsampled point $\mathbf{x}_i^U$ is not determined for this time. The propagation process is repeated until all points are segmented, as described in Algorithm 2. This is denoted as NN+PP in the following sections.

4. Experiments

4.1. Dataset: Shapenet

ShapeNet [17] is a richly-annotated, large-scale dataset of 3D shapes that covers 55 common object categories. It provides 3D models that can be utilized for various tasks (e.g., 3D shape classification, part segmentation, and semantic segmentation). The ShapeNet part segmentation dataset is a subset of the ShapeNet designed for part segmentation of 3D objects. It consists of 16,881 3D objects obtained from 16 categories of common things, annotated with 50 parts in total. Each object has 2 to 5 distinct parts. For the PCS experiments, point clouds are sampled from the 3D objects according to the settings in [19]; point clouds are composed of 2620 points on average. The train test split follows [17].

4.2. Neural network: PointNet++

PCS Model. The ultimate PCS performance was highly dependent on the performance of NN. We utilized PointNet++ [12], an end-to-end deep learning framework specialized for point clouds.

Algorithm 2 PCS via single inference using NN with sampling for the initial sampled points and then via iterative probability propagation for the other points (**Ours**, **NN+PP**).

```

1: Input: the coordinates (x, y, z) of a point  $\mathbf{x}$  all the points of a
   point cloud  $\mathbf{X}$ , the unlabeled points  $\mathbf{X}^U$ , the number of points
   for the model input  $N^S$ , the probability distributions  $p_{NN}$ , the ac-
   cumulated probability distribution  $p_{PP}$ , and the effective distance
   ratio  $d_E$ 
2: Output: the predicted results  $\mathbf{y}$  for all the points  $\mathbf{X}$ 
3:  $\mathbf{X}^S \leftarrow$  the initial sampled  $N^S$  points from  $\mathbf{X}^U$ 
4:  $\mathbf{X}^U \leftarrow \{\mathbf{x} \in \mathbf{X}^S\}$ 
5:  $\mathbf{y}^S \leftarrow p_{NN}(\mathbf{y}^S | \mathbf{X}^S)$  as in Eq. (2)
6:  $d_{E, default} \leftarrow d_E$ 
7: while  $length(\mathbf{X}^U) > 0$  do
8:    $\mathbf{X}^A \leftarrow \{\}$ 
9:   for  $\mathbf{x}_i^U$  in  $\mathbf{X}^U$  do
10:     $\mathbf{y}_i^U \leftarrow p_{PP}(\mathbf{y}_i^U | \mathbf{x}_i^U; \mathbf{x}_i^U \in \mathbf{X}^U, p(\mathbf{y}^S | \mathbf{X}^S))$ 
       as in Eq. (6)
11:    if  $\mathbf{y}_i^U \neq 0$  then
12:       $\mathbf{X}^A \leftarrow \mathbf{X}^A \cup \{\mathbf{x}_i^U\}$ 
13:       $\mathbf{y}^S \leftarrow \mathbf{y}^S \cup \{\mathbf{y}_i^U\}$ 
14:    end if
15:  end for
16:   $\mathbf{X}^S \leftarrow \mathbf{X}^S \cup \mathbf{X}^A$ 
17:   $\mathbf{X}^U \leftarrow \mathbf{X}^U - \mathbf{X}^A$ 
18:   $\mathbf{y} \leftarrow \mathbf{y} \cup \mathbf{y}^S$ 
19:  if  $length(\mathbf{X}^A) > 0$  then
20:     $d_E \leftarrow d_{E, default}$ 
21:  else
22:     $d_E \leftarrow d_E \times 1.5$ 
23:  end if
24: end while

```

It consists of a sampling and grouping layer for building a hierarchical grouping of points and PointNet [11] layer, which includes max-pooling and fully-connected layers to extract features of larger regions along the hierarchy progressively. The extracted features are subsequently aggregated to extract global features with multi-resolution grouping methods. In this study, the model was trained with a batch size of 16 and an epoch of 300. Early stopping was set with the patience of 50. Adam optimizer was utilized with a learning rate of 0.001, betas ranging from 0.9 to 0.999, and a decay rate of 0.0001. It should be noted that the training, verification, and testing were conducted on the sampled points according to each sampling condition. The best performance models were used for the performance comparison between using NN alone and using NN+PP.

Sampling Methods. We sampled the input point cloud into subsets of 128, 256, 512, 1024, and 2048 points using three different sampling methods: random sampling, Poisson disk sampling [20], and farthest point sampling [12]. The number of sampled points, 128, 256, 512, 1024, and 2048, corresponded approximately 5%, 10%, 20%, 39%, and 78% of the averaged total number of points, respectively. The random sampling method selects subsets of points from all the sets with the same probability of selection for each point. While random sampling provides a uniform probability of sample placement, it does not ensure that the sampled points are distributed uniformly. The Poisson disk sampling selects points with the constraint that no two selected samples are closer than the Poisson disk radius. When compared to random sampling, the Poisson disk sampling produces more uniformly distributed samples across the sampling domain due to the constraints of distance among the selected points, as illustrated in Fig. 2. The farthest point sampling is the most widely utilized method that

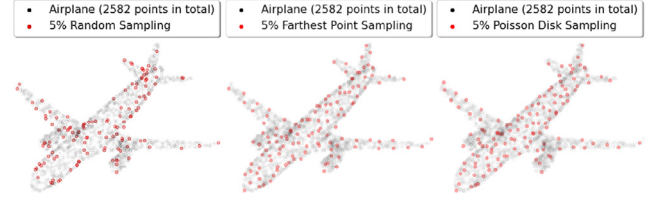


Fig. 2. The visualization of the sampled points for the sampling method with a sampling ratio of 5%.

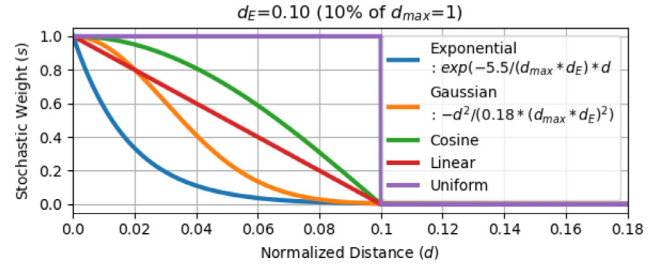


Fig. 3. Uniform, cosine, linear, Gaussian, and exponential distributions for probability propagation with the effective distance ratio (d_E : 0.1).

selects a group of points farthest apart from each other. Through the iterations, the subsets of points are selected by maximizing the distance between them. Similar to the Poisson disk sampling method, farthest point sampling has better coverage of the sampling domain than the random sampling method.

4.3. Probability propagation settings

As shown in Fig. 3, five probability distributions (uniform, cosine, linear, Gaussian, and exponential distributions) were used as the stochastic weight functions, $S(\cdot)$, and set effective distance ratios, d_E , ranging from 2% to 80% in 2%p increments. Optimal combinations of propagation weight function and effective distance ratio were investigated through ablation studies.

4.4. Metrics

Instance mIoU was used as a PCS performance measure. The mIoU is calculated by taking the IoU, a.k.a. Jaccard index, of each class or each instance and averaging them. The IoU is the number of points where the predicted label and the ground truth are the same divided by the number of points where the predicted label and the ground truth exist. Relative inference time and average iteration count are compared in terms of efficiency. The performance was measured under the same conditions using CPU (AMD Ryzen 9 3900X) and GPU (NVIDIA GeForce RTX 3070) in Ubuntu 18.04.6 LTS.

5. Results and discussion

The best-trained NN for each sampling condition was used to compare the PCS performance between using sampling and NN for all points and using NN for the initial sampled points and iterative PP for the other points. For the sampling methods (random, farthest point, and Poisson disk sampling), sampling ratio (about 5%, 10%, 20%, 39%, and 78% of the average total number of points in the point cloud), and effective distance ratio (from 2% to 80% in 2%p increments), the instance mIoU is shown in Fig. 4.

NN with Poisson disk sampling achieved the highest instance mIoU compared with NN with farthest point sampling and random sampling. Regardless of the sampling method, the instance mIoU with NN+PP was always higher than with NN alone. These results

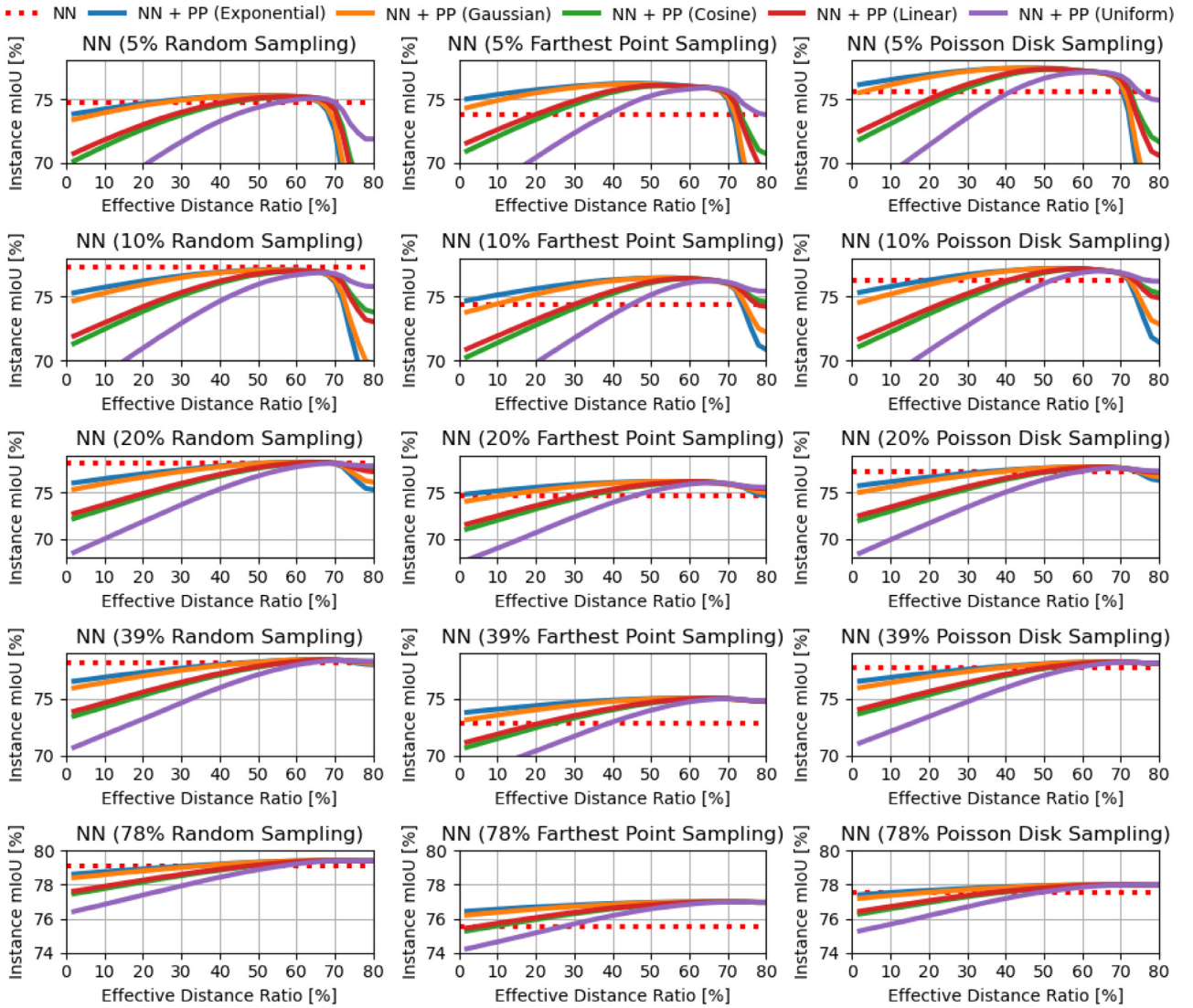


Fig. 4. Instance mean intersection over union (mIoU) for sampling method, sampling ratio, and effective distance ratio.

support the claim that PP contributes to PCS performance improvement and more physically reasonable segmentation based on the number of neighboring points and their proximity by spreading the probability distribution of each point to geometrically adjacent points. Unlike the farthest point and Poisson disk sampling methods, random sampling does not ensure that sampled points cover the point cloud regions evenly, especially when the density of points is imbalanced. The normalized distances between the closest neighboring points have different standard deviations according to the sampling method, as shown in Fig. 5. The probability density of the random sampling is more dispersed than that of the other sampling methods in the histogram. The points sampled by random sampling were more locally clustered than those by the other sampling methods, and this phenomenon appeared to be obvious as the sampling ratio was small. These results suggest that the sampling method with a low standard deviation of the distances between the closest neighboring points is appropriate for PP.

Exponential, Gaussian, cosine, linear, and uniform distribution are arranged in descending order of their performance when using NN+PP. This means that it is advantageous to use PP to assign higher probabilistic weight to closer and adjacent points. PP demonstrated a better effect with fewer sampling points, implying

that PP is useful for conditions requiring high efficiency. The performance increased until the best effective distance ratio for each sampling condition, and then it decreased thereafter. This is because the influence of the probability distributions of irrelevant points increases as the effective distance gets longer. This characteristic was more apparent in using the exponential distribution compared to those of the other distributions. When using the exponential distribution, the stochastic weight increases much more drastically as the distance to the adjacent point decreases, i.e., the neighboring point is closer, as shown in Fig. 3. This is attributed to the fact that negative probability distributions of irrelevant points spread as strongly as positive probability distributions of relevant points spread strongly. In particular, a small effective distance ratio is preferable to improve the efficiency when using PP since the computational cost to accumulate the probability distributions of close neighboring points increases exponentially. Therefore, the exponential distribution is recommended for PP because of its superior performance and a wider range of the optimal effective distance, i.e., it is favorable to choose a small effective distance ratio for high efficiency with reasonable performance.

Furthermore, we performed a class-wise mIoU comparison between NN and NN+PP in the best condition using NN with Poisson disk sampling and PP based on exponential distribution. As

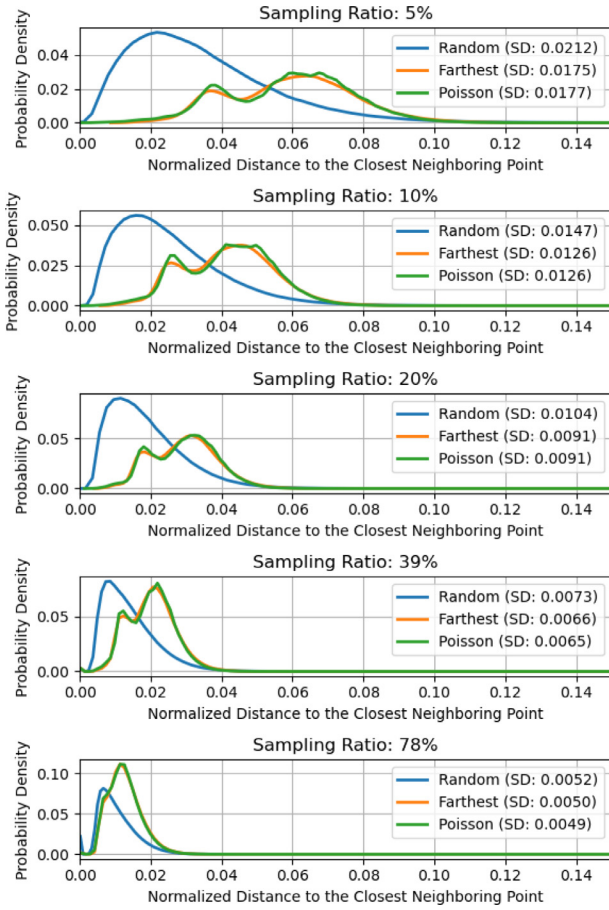


Fig. 5. Normalized distance to the closest neighboring point according to the sampling condition (SD: standard deviation).

shown in Table 1, the results indicate that NN+PP achieves a performance level comparable to NN for almost all categories with reduced time. Notably, for categories such as Lamp and Earphone, NN+PP outperformed NN by a significant margin, demonstrating its effectiveness.

The efficiency of PP based on the exponential distribution was investigated for effective distance ratios by comparing the relative inference time and average iteration count, as shown in Supplementary Figs. 2–4. The performance comparison results according to the sampling conditions are shown in Fig. 4. The inference time required to segment a single object is shown in Table 2. The results demonstrate that NN+PP achieves higher instance mIoU and requires less inference time than NN alone. Using PP based on an exponential distribution significantly improves PCS performance and efficiency.

When using the exponential distribution for PP, the recommended effective distances according to the sampling condition (sampling method and sampling ratio) for NN are shown in Fig. 6. For a low sampling ratio, a long effective distance ratio is recommended for the best performance; however, for a large sampling ratio, a short effective distance ratio is favorable for the best performance. When using Poisson disk sampling for NN and exponential distribution with the best effective distance for PP, both the performance and efficiency in using NN+PP were superior to those in using NN alone as shown in Fig. 7. Particularly, as the sampling ratio decreases, PP significantly reduced the inference time, i.e., dramatically improved the efficiency. When the number of sampled points was less than 50% of the average total number of points, PP considerably speeded up the PCS, i.e., remarkably im-

Table 1
Best class-wise mean intersection over union of using NN or NN+PP with Poisson disk sampling and exponential distribution.

Sampling Ratio	Method	Best Class-wise Mean Intersection over Union (bold text indicates higher performance value)															Average
		Cap	Rocket	Skateboard	Mug	Laptop	Car	Pistol	Guitar	Airplane	Chair	Motorbike	Table	Knife	Lamp	Earphone	
5%	NN	60.61	38.95	68.92	80.55	95.07	60.26	74.16	84.22	62.85	84.71	30.21	77.52	77.90	72.98	52.17	
	NN+PP	57.46	36.78	68.07	79.98	93.76	59.11	73.66	84.13	62.86	86.49	31.54	79.50	80.81	80.50	69.04	
10%	NN	60.82	46.22	68.10	78.09	95.18	55.69	71.86	87.80	61.98	86.85	25.19	77.56	80.58	75.47	63.22	
	NN+PP	55.95	44.67	68.49	77.20	94.70	54.33	70.82	87.81	62.29	87.21	28.30	79.35	82.66	79.20	68.41	
20%	NN	57.96	56.62	71.21	79.62	95.25	58.07	77.42	87.13	63.20	86.57	30.27	78.64	80.90	77.41	68.22	
	NN+PP	56.57	55.64	69.90	79.67	94.82	57.63	77.28	87.20	63.20	86.90	30.73	79.44	81.79	79.58	69.69	
39%	NN	61.59	49.92	69.95	83.10	94.54	58.93	75.97	88.08	65.94	87.08	35.80	78.62	80.12	78.58	73.21	
	NN+PP	60.69	49.25	68.97	81.74	94.17	59.06	75.81	87.98	66.17	87.95	36.39	79.55	80.36	79.24	73.93	
78%	NN	66.04	50.53	69.40	88.67	94.94	56.96	73.81	86.23	68.03	86.63	33.79	77.68	77.69	78.14	67.12	
	NN+PP	66.13	50.31	69.09	88.45	94.94	57.54	73.78	86.09	68.51	87.17	33.87	78.26	77.85	78.64	67.11	
																	78.00

Table 2

Inference time per object at the best instance mean intersection over union according to sampling condition (unit: second).

Sampling Method	Inference Method	Sampling Ratio				
		5%	10%	20%	39%	78%
Random Sampling	NN	4.881	2.339	1.238	0.504	0.287
	NN+PP	0.153	0.141	0.203	0.243	0.264
	NN/(NN+PP)	31.810	16.641	6.112	2.079	1.087
Farthest Point Sampling	NN	4.661	2.576	1.453	0.796	0.451
	NN+PP	0.046	0.070	0.134	0.198	0.206
	NN/(NN+PP)	102.267	36.649	10.875	4.016	2.184
Poisson Disk Sampling	NN	2.769	1.171	0.692	0.329	0.208
	NN+PP	0.048	0.071	0.110	0.171	0.225
	NN/(NN+PP)	57.659	16.415	6.311	1.927	0.923

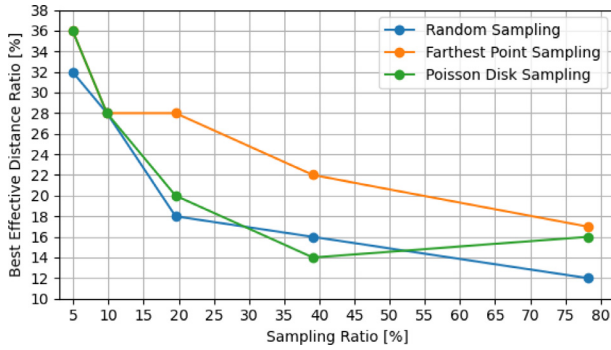


Fig. 6. Best effective distance ratio for sampling method and its ratio at the best instance mean intersection over union.

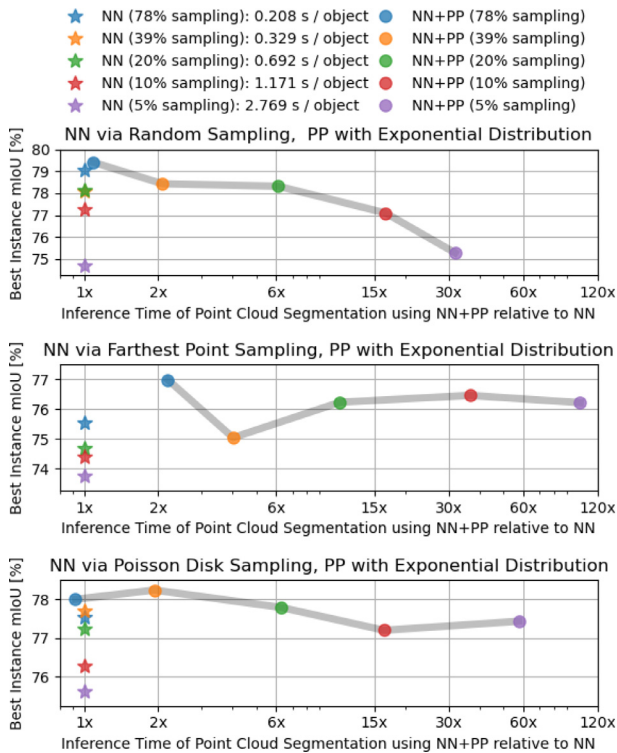


Fig. 7. Inference time of point cloud segmentation using NN+PP with exponential distribution relative to using NN at the best instance mean intersection over union (mIoU).

proved efficiency. Especially when using 5% farthest point sampling for NN, the instance mIoU with NN+PP was improved by approximately 2.457p, and the inference speed is 102 times faster than using NN alone.

6. Conclusions

We proposed *Probability Propagation* (PP) as a stochastic mapping method for faster and more efficient PCS with better performance while reducing memory usage and inference time. Empirically, we investigated the efficiency of PP in various settings: sampling methods, sampling ratios, and effective distance ratios. Regardless of the sampling methods, the instance mIoU with NN+PP showed higher performance, and the inference time was much shorter than when using NN only. Specifically, PP based on exponential distribution was superior to others regardless of the performance metric, thereby indicating that it is advantageous to use PP to assign a higher probabilistic weight to close adjacent points. Especially, adopting PP yielded better results in terms of fewer sampling points; when the number of sampled points was less than 50% of the average total number of points, PP significantly reduced the inference time, i.e., dramatically improved the efficiency. Specifically, using farthest point sampling with the sampling ratio of 5%, NN+PP improved the instance mIoU by 2.457p with 102 times faster PCS compared to that when using NN alone. This indicates that PP is more effective for harsh conditions requiring high efficiency and low battery usage. Furthermore, the proposed method can be easily applied to other frameworks for PCS or image segmentation. Consequently, PP can replace the part of the NN inferences with enhanced simplicity and efficiency, which is the more friendly approach for constrained situations such as mobile robots and self-driving drones. Significant improvements in performance and efficiency when using NN+PP indicate that PP enables the segmentation of numerous points measured using laser scanners on edge AI systems that require low-cost computing costs with limited memory. Especially outdoor mobile systems such as mobile robots, autonomous automobiles, and unmanned aerial vehicles need to continuously collect fine point clouds and grasp the surrounding situation in real time based on the accumulated point clouds. Repetition of point sampling and inference using NN alone is time-consuming and requires expensive computational costs that accelerate battery discharge and significantly reduce device up-time. In addition, a long inference time causes delays in decision-making based on the situation, making it difficult to respond to the rapidly changing environment properly. Since this can cause fatal accidents and device damage, mobile systems require fast and efficient PCS. To this end, the proposed PP is an effective and powerful solution for the segmentation of the large-scale point cloud with high-speed and low-cost computing as well as performance

enhancements. Therefore, the adoption of PP can greatly expand the applicability of PCS to edge AI systems.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Hogeon Seo reports financial support was provided by Korea Atomic Energy Research Institute. Kyoobin Lee reports financial support was provided by Institute for Information Communication Technology Planning and Evaluation.

CRediT authorship contribution statement

Hogeon Seo: Conceptualization, Methodology, Software.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was supported by a grant from the [Korea Atomic Energy Research Institute R&D Program](#) (No. KAERI-524450-23) as well as by a grant from the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korean government (MSIT) (No. 2020-0-00857, Development of cloud robot intelligence augmentation, sharing and framework technology to integrate and enhance the intelligence of multiple robots).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.patrec.2023.04.010](https://doi.org/10.1016/j.patrec.2023.04.010).

References

- [1] R. Kaijaluoto, A. Kukko, A. El Issaoui, J. Hyypää, H. Kaartinen, Semantic segmentation of point cloud data using raw laser scanner measurements and deep neural networks, *ISPRS Open J. Photogramm. Remote Sens.* (2021) 100011.
- [2] A. Nguyen, B. Le, 3D point cloud segmentation: a survey, in: 2013 6th IEEE Conference on Robotics, Automation and Mechatronics, IEEE, 2013, pp. 225–230.
- [3] A. Nurunnabi, G. West, D. Belton, Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data, *Pattern Recognit.* 48 (4) (2015) 1404–1419.
- [4] J.M. Biosca, J.L. Lerma, Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods, *ISPRS J. Photogramm. Remote Sens.* 63 (1) (2008) 84–98.
- [5] H. Zhou, Z. Fang, Y. Gao, B. Huang, C. Zhong, R. Shang, Feature fusion network based on attention mechanism for 3D semantic segmentation of point clouds, *Pattern Recognit. Lett.* 133 (2020) 327–333.
- [6] M. Zhang, P. Kadam, S. Liu, C.C.J. Kuo, GSIP: green semantic segmentation of large-scale indoor point clouds, *Pattern Recognit. Lett.* 164 (2022) 9–15.
- [7] N. Luo, Y. Wang, Y. Gao, Y. Tian, Q. Wang, C. Jing, kNN-based feature learning network for semantic segmentation of point cloud data, *Pattern Recognit. Lett.* 152 (2021) 365–371.
- [8] N. Zhao, T.-S. Chua, G.H. Lee, Few-shot 3D point cloud semantic segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8873–8882.
- [9] D. Maturana, S. Scherer, VoxNet: a 3D convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2015, pp. 922–928.
- [10] L. Chapmi, C. Choy, I. Armeni, J. Gwak, S. Savarese, SEGCloud: semantic segmentation of 3D point clouds, in: 2017 International Conference on 3D Vision, IEEE, 2017, pp. 537–547.
- [11] C.R. Qi, H. Su, K. Mo, 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, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc., 2017.
- [13] H. Zhao, L. Jiang, J. Jia, P.H.S. Torr, V. Koltun, Point transformer, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16259–16268.
- [14] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, A. Trigi, A. Markham, RandLA-Net: efficient semantic segmentation of large-scale point clouds, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11105–11114.
- [15] A.-V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, *ISPRS J. Photogramm. Remote Sens.* 104 (2015) 88–100.
- [16] R.A. Rosu, P. Schütt, J. Quenzel, S. Behnke, LatticeNet: fast point cloud segmentation using permutohedral lattices, in: *Proc. of Robotics: Science and Systems (RSS)*, 2020.
- [17] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., ShapeNet: an information-rich 3D model repository, *arXiv preprint arXiv:1512.03012* (2015).
- [18] P.K.D. Pramanik, N. Sinhababu, B. Mukherjee, S. Padmanaban, A. Maity, B.K. Upadhyaya, J.B. Holm-Nielsen, P. Choudhury, Power consumption analysis, measurement, management, and issues: a state-of-the-art review of smartphone battery and energy usage, *IEEE Access* 7 (2019) 182113–182172, doi:10.1109/ACCESS.2019.2958684.
- [19] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, A scalable active framework for region annotation in 3D shape collections, *SIGGRAPH Asia* (2016).
- [20] C. Yuksel, Sample elimination for generating Poisson disk sample sets, in: *Computer Graphics Forum*, Vol. 34, Wiley Online Library, 2015, pp. 25–32.