# AAG-Net: Angle-Attention Graph Neural Network for Point Cloud Sampling

Seokjin Hong, Jongwon Jeon, and Jinwoo Yoo, *Member, IEEE*

*Abstract*— **Lidar sensors are essential in autonomous driving, providing accurate 3D scans of the vehicle's surroundings and generating point cloud data crucial for detecting and tracking road conditions, vehicles, and pedestrians. However, lidar-generated point clouds are often vast, making real-time processing and analysis challenging. Thus, efficient data sampling methods are necessary. Traditional non-learning methods, such as farthest point sampling (FPS), can sample uniformly, but have limitations in generating sample points that are optimized for a specific task. Therefore, recent research has focused on task-specific point sampling methods using deep learning. However, while recent deep learning-based sampling methods exploit the location and feature differences of neighboring points to learn local features of a point cloud, their ability to effectively exploit the geometric information of closely spaced points is limited. To address this issue, we propose AAG-Net, a framework for point cloud sampling that is optimized for classification tasks by exploiting the geometric features of point clouds, including their angular characteristics. The framework introduces an attention mechanism that exploits the angles of neighboring points and designs a Global-Local Graph Point Attention (GL-GPA) scheme to effectively integrate global and local features. Experimental results on the ModelNet40 dataset show that the proposed method significantly improves point cloud sampling performance compared to state-of-the-art methods.**

*Index Terms*— **Attention mechanism, deep learning, graph neural network, LIDAR point cloud sampling.**

## I. INTRODUCTION

POINT clouds, a data format consisting of a set of points in 3D space, are actively researched in various fields such as autonomous driving, robotics, and computer vision because they can accurately model real-world objects or environments in 3D. Point clouds are also used for various tasks such as object classification and semantic segmentation, allowing accurate recognition and analysis of complex environments [1], [2], [3], [4]. Recently, deep learning research has been actively conducted on large-scale point clouds acquired by LiDAR [5], [6], [7]. However, in order to use point clouds for deep learning, large-scale point cloud data must be processed

effectively. Therefore, research on sampling that reduces the number of points in the original data becomes important. Such sampling minimizes storage and computational costs by reducing the amount of data while preserving important properties. It also reduces processing time by increasing the efficiency of deep learning models.

Point cloud sampling methods can be divided into two types: Task agnostic methods include farthest point sampling (FPS) and random sampling (RS). These methods work without being influenced by downstream tasks. However, they can lead to inefficient performance because they do not generate the best points for a particular task. Task specific methods include deep learning-based methods such as S-NET [8], and, SampleNet [9], [10], [11]. They have been studied to improve sampling performance by generating optimized points guided by a pre-trained downstream network. However, there are two problems with existing methods. First, S-NET [8] and SampleNet [9] use PointNet [1] to extract point features, which does not fully reflect the local structural features that contain fine-grained geometric information. Second, the location and feature differences between points alone do not fully reflect the geometric information, which limits their ability to handle dense points within a region.

In this letter, we propose AAG-Net with two proposed modules to sample by providing additional geometric information in the local coverage to solve the above problems. We also construct a hierarchical neural network to learn local features in different coverage.

Our main contributions can be summarized as follows.

- We introduced an attention mechanism that uses the angles of neighboring points to selectively focus on the most relevant points in the graph neighborhood.
- For object structure-based sampling, we integrated objects' global features with detailed local information by connecting global features in the point cloud with local features captured hierarchically in small neighborhoods.

## II. METHODS

The proposed AAG-Net is mainly related to several existing works, including GAC [14] and PointNet++ [2]. We build on the idea of the attention mechanism in GAC [14] and propose a new attention mechanism that utilizes angular information.

Seokjin Hong and Jongwon Jeon are with the Graduate School of Automobile and Mobility, Kookmin University, Seoul 02707, Republic of Korea (e-mail: cheongsu030536@gmail.com).

Jinwoo Yoo is with the Department of Automobile and IT Convergence, Kookmin University, Seoul 02707, Republic of Korea (e-mail: jwyoo@kookmin.ac.kr).
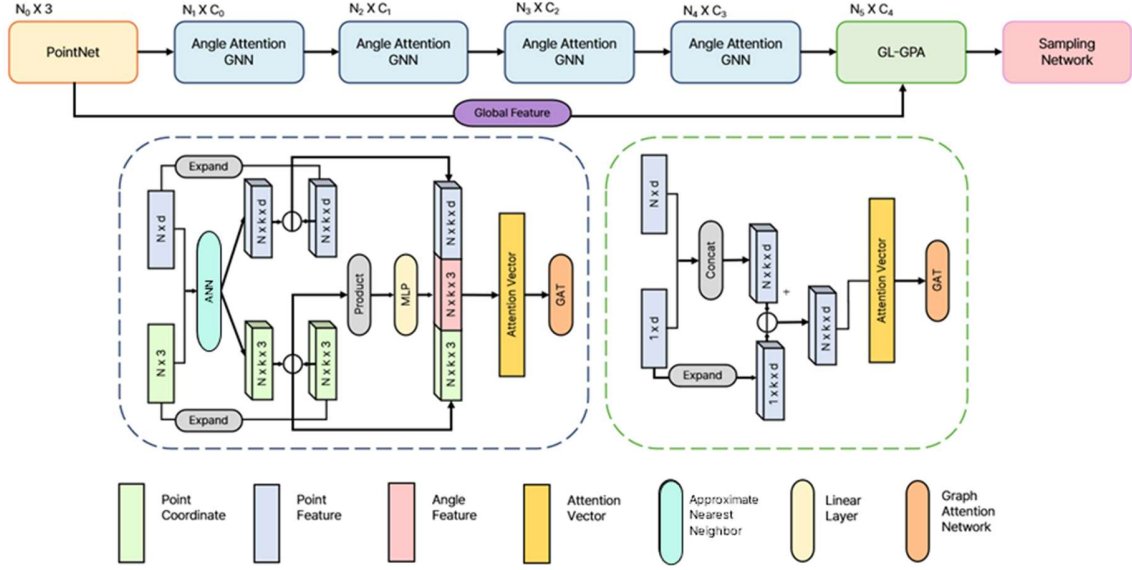
Fig. 1. Overall architecture of AAG-Net, based on hierarchical learning, consisting of two parts: (a) Angle Attention GNN to extract local features based on angles, and (b) GL-GPA to integrate global and local features.

The following describes a framework for sampling classification-oriented 3D point clouds. An overview of our proposed AAG-Net is shown in Fig. 1, which includes three main components: first, PointNet [1] embeds the features of the point cloud and obtains global information through the max-pooling operation. Second, it assigns weights to the nearest neighboring points and generates downsampled points through a hierarchical structure. Finally, the previously obtained global and local feature points are connected by a graph to learn and create a unified feature. The following sections describe the framework's main architectural design.

*A. Angle Attention Mechanism*

The input point cloud is processed through four successive layers of AAG. First, we build a graph of the point features extracted via PointNet [1] to use as input to the AAG. We take an input point cloud $P = \{p_1, p_2, \dots, p_n\} \in R^3$, denoted by $p_i = [x_i, y_i, z_i]$, which represents the 3D coordinates of the $i$-th point. From the coordinates and features $H = \{h_1, h_2, \dots, h_n\} \in R^F$, previous methods normally utilize k-NN for neighbor selection. However, due to scalability concerns arising from the high-dimensional nature of the features and increased dataset size, we adopted approximate nearest neighbor (ANN) search, specifically using the ANNOY(Approximate Nearest Neighbors Oh Yeah) algorithm, to efficiently select neighbors. For each center point $i$, we compute the attentional weight by utilizing the position and feature differences of the neighboring point set $N(i)$ and the angle $\theta$ of the relative position vector.

$$a_{ij} = \alpha \left( \Delta p_{ij}, \Delta h_{ij}, \varphi(\Delta \theta_{ij}) \right), j \in N(i) \qquad (1)$$

$a_{ij}$ represents the attentional weight of center point $i$ and neighbor point $j$. $\Delta p_{ij}$ is the relative position, and the 3D coordinate position of the neighboring points relative to the center point is used to grasp the spatial structure to obtain local information. $\Delta h_{ij}$ is the relative feature, and the feature difference and angle between points are used as the input of $\alpha$ to generate the attentional weight so that the center point can recognize the local structure and aggregate more important neighboring points. $\alpha$ and $\varphi$ are MLPs. Specifically, rather than directly concatenating raw angle values to the attention mechanism, an MLP (Angle MLP) is used to preprocess and embed these angular values into more expressive representations. This allows the attention mechanism to better capture and leverage meaningful geometric relationships, enhancing the overall representational capacity of the model.

$$a_{ij} = \frac{exp(a_{ij})}{\Sigma_{j \in N(i)} \exp(a_{ij})} \qquad (2)$$

The attentional weights of center point $i$ and neighboring point $j$ are normalized with respect to their neighbors. To normalize, the attentional weight for neighboring point $j$ is calculated by applying softmax to it.

$$h_i^{(l)} = \sigma \left( \sum_{j \in N(i)} a_{ij} w^{(l)} h_j^{(l-1)} \right) \qquad (3)$$

Thus, the embedding of the points can be formulated as (3), where $w^{(l)}$ is the learning parameter corresponding to the $l$-th layer and $\sigma$ is the activation function.

The main idea of the module is to include the angles between points in the attention calculation to provide additional information about the spatial relationship between the center point and its neighbors. When constructing graphs using ANN, points in the same direction tend to provide similar location
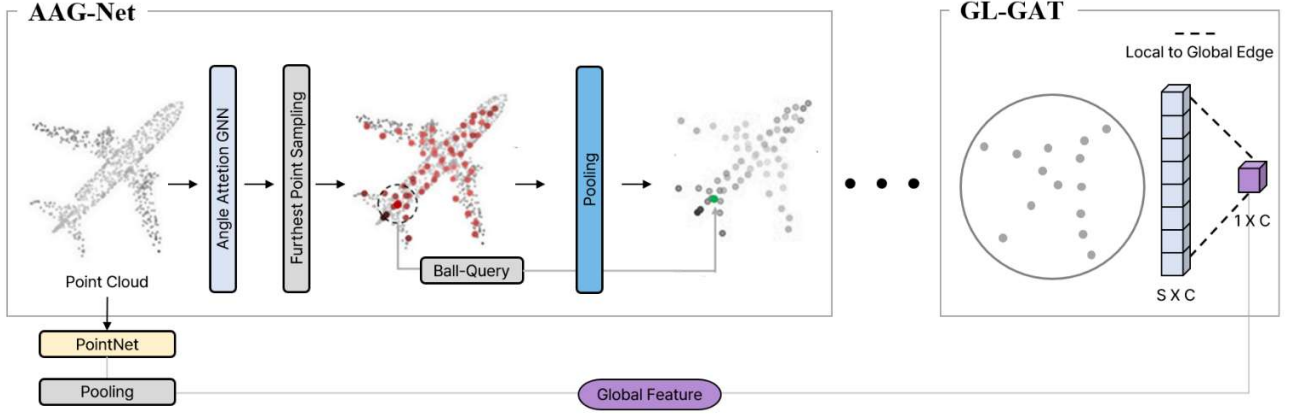
Fig. 2. Overview of GL-GPA method. After selecting points with FPS, a ball query is used to pool points within a radius. After iterating, the local and global features are connected by edges to get the final features through GAT.

information. Therefore, when calculating the attentional weight, the angle between points is used to better understand the directionality of the point distribution to learn its neighbors.

$$\theta = \left\{ \left( \theta_{xi}, \theta_{yi}, \theta_{zi} \right) \in R^3 \right\}_{i=1}^{k} \qquad (4)$$

$\theta_{xi}, \theta_{yi}, and\ \theta_{zi}$ denote the angles on the $x$, $y$, and $z$ axes of the k neighbors from the center point. In our methodology, we take center point $p_i$ and its neighbors $p_j$ between a given center point and its neighbors. This vector is used to calculate the angles by taking the dot product with the 3D axes, as shown in the following equation:

$$\theta_{ij,x} = \arccos \left( \frac{\Delta p_{ij} \cdot ex}{\|\Delta p_{ij}\|} \right)$$
$$\theta_{ij,y} = \arccos \left( \frac{\Delta p_{ij} \cdot ey}{\|\Delta p_{ij}\|} \right) \qquad (5)$$
$$\theta_{ij,z} = \arccos \left( \frac{\Delta p_{ij} \cdot ez}{\|\Delta p_{ij}\|} \right)$$

where $\Delta p_{ij}$ represents the relative position vector between two points, $ex, ey, and\ ez$ represent the $x$, $y$, and $z$ axes, respectively, and $\Delta \theta_{ij}$ represents the angle between the center point $p_i$ and neighbor points $p_j$. As shown in Fig. 3, the three angles of the neighbor point can be obtained.

This allows way in which the features of each neighboring point are reflected in the context of the center point to be controlled. The angle attention mechanism provides a more precise understanding of the relative position between the center point and its neighbors, calculating an attentional weight that also considers the spatial orientation of the neighbors.

*B. Global-Local Graph Point Attention*

In this section, we adopt the hierarchical point set feature learning process proposed by PointNet++ [2] as the basis for hierarchical learning of point clouds. At each layer, we use the
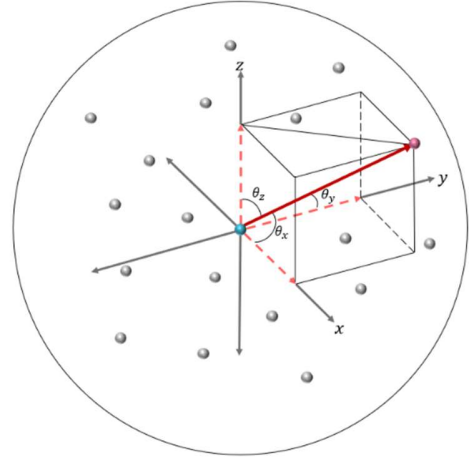


Fig. 3. Three angles between center point and its neighbors.

farthest point sampling (FPS) method to select and pool points, reducing the number of points hierarchically while preserving the spatial structure more effectively. The FPS method simplifies the data by selecting the farthest points at each layer, thus better preserving the spatial structure. In this section, the points containing hierarchically learned spatial structure information are connected to global information feature vectors and graphs to combine local and global information. As shown in Fig. 2, 1CONV (Convolution), a learning method of PointNet [1], is applied to the input point cloud $N_0 \times 3$ to generate a feature vector of $N \times C$ dimension. Then, the global information feature dimension obtained by applying max-pooling is $1 \times C$. To obtain local information features, the AAG layer is repeatedly performed to generate local information features of $S \times C$. $S$ is the number of points sampled by the FPS algorithm. By connecting the global and local information with directed edges from each local point to the global point, we construct a directed graph consisting of $(S + 1) \times C$ point features and $2 \times S$ directed edges. Each local point connects individually to the global point, establishing a unidirectional relationship from local to global features. Finally, GAT [39] is used to learn to integrate the global and local information.

*C. Loss Functions*

The training of the point cloud sampling network reflects the error information of the classification network and the error information based on the distance between the sampling points and the original points. Task loss $L_{task}$ uses a frozen classification network and a cross-entropy loss function to minimize the difference between the model output and the actual label. The sampling loss $L_{sample}$ aims to minimize the distance between the sampled points and the original point cloud, which consists of two components: average nearest neighbor loss $L_a$, which increases sampling consistency by minimizing the distance between sampled points, and maximum nearest neighbor loss $L_m$, which improves sampling quality by minimizing the maximum distance between sampled points and the original point cloud. Additionally, an extra term is included in the sampling loss to ensure that the sampled points are evenly distributed within the original point cloud. Given $P$ as the input point cloud and $Q$ as the sampled points, the equation is

$$L_{\text{sample}}(Q, P) = L_a(Q, P) + \beta L_m(Q, P) + (\gamma + \delta|Q|)L_a(P, Q)$$
$$(6)$$

where $\beta$, $\gamma$, and $\delta$ are hyperparameters that adjust the size between losses. The total loss $L_{total}$ is a weighted sum of the working loss and sampling loss, with a hyperparameter $\lambda$ that balances the two losses:

$$L_{total} = L_{\text{task}} + \lambda L_{\text{sample}}(Q, P) \qquad (7)$$

## III. EXPERIMENTAL RESULTS

*A. Datasets*

To evaluate the performance of the proposed approach on the 3D point cloud sampling task, we utilized the ModelNet40 dataset, which consists of computer-generated 3D CAD models. The dataset contains a total of 12,311 instances categorized into 40 different classes, with carefully labeled models standardized in size and orientation. The ModelNet40 dataset is provided with a predefined split consisting of 9,843 instances for training and 2,468 instances for testing, following official train-test split. ModelNet40 is widely used as a benchmark for evaluating the performance of classification models on point cloud data.

*B. Implementation Details*

For the implementation of the network used Python 3.7 and PyTorch 1.9.1 were used. Model training was conducted on a system with an NVIDIA RTX 3090ti equipped with 24GB of memory and CUDA 11.1. During training, stochastic gradient descent was used, along with a cosine annealing scheduler without warm restarts. The loss function used was cross-entropy with label smoothing. The initial learning rate was set to 0.01. The hyperparameters β, γ, and δ were empirically set based on values provided by SampleNet [Ref] (β=1, γ=1, δ=0),

and the hyperparameter λ=30 was empirically determined through preliminary experiments to optimally balance the task-specific and sampling losses. Training was performed using 1024 points, with a batch size of 24, over 300 epochs.

*C. Results and Analysis*

TABLE I
CLASSIFICATION ACCURACY (%) ON MODELNET40

| Sample size | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| RS | 8.7 | 24.87 | 54.53 | 79.26 |
| FPS | 24.31 | 55.12 | 76.92 | 87.53 |
| SampleNet | 78.36 | 80.60 | 80.32 | 79.36 |
| APSNet | 81.42 | 83.89 | 88.15 | 88.38 |
| AAG-Net(ours) | 87.24 | 87.93 | 88.25 | 88.70 |

This section provides an evaluation of the proposed AAG-Net sampling network against existing methods. AAG-Net outperformed existing sampling methods in experiments based on the ModelNet40 [13] dataset. We evaluated different sampling methods and numbers of points, and like other sampling networks, we used PointNet [1] as the classification task network. PointNet [1] achieves 90.1% accuracy using 1,024 original point clouds, and the performance of the sampling network was evaluated based on this criterion. The results of the experiments with 1,024 point clouds sampled with 8, 16, 32, and 64 points are presented in Table I. AAG-Net showed consistently high performance at all sampling sizes.

TABLE II
PRECISION(%) AND RECALL (%) ON MODELNET40

| Sample size | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| Precision | 8.7 | 24.87 | 54.53 | 79.26 |
| Recall | 24.31 | 55.12 | 76.92 | 87.53 |

The proposed approach demonstrated high performance even with small sampling sizes by effectively utilizing angle information. This was achieved by better understanding the positional distribution of the original point cloud and efficiently representing complex shapes with a smaller number of points. Fig. 4 visualizes the sampled point cloud from the input point cloud, where the gray points represent the input points and the green points represent the sampled points. Ideally, point cloud sampling aims to uniformly sample the original point cloud. However, in practice, the sampling network samples points optimized for the classification task, indicating that it selects specific types of points preferred by the classification network.

To test the effectiveness of the angle attention mechanism, we conducted a comparison experiment between learning position and feature differences with and without angles. To compare the two methods, we calculated the attention scores to understand the impact of angles on the attention score of
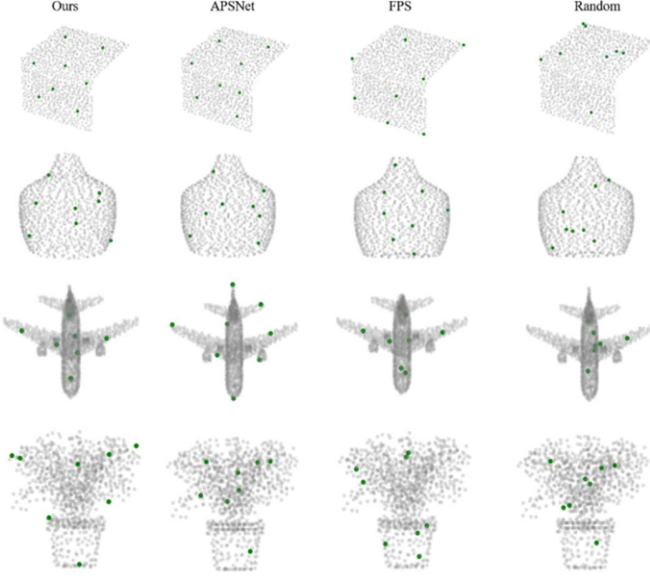
Fig. 4. Visual comparison of point cloud sampling on ModelNet40 dataset. Compared to the task-agnostic method, the task-specific method produces optimized points for classification.
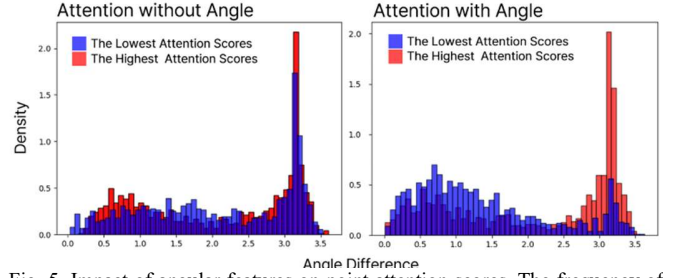


Fig. 5. Impact of angular features on point attention scores. The frequency of angular differences between neighboring points is visualized in the graphs, where blue indicates the angular difference between the two points with the lowest attention scores among neighbors and red indicates the angular difference between the two points with the highest attention scores.

helps better understand and utilize the spatial structure of the data, demonstrating that using angle information along with the attention mechanism is an effective approach.

neighboring points. We calculated the difference in angles between the two point pairs with the highest and lowest attentional weights in a range of neighboring points. We measured the number of times each pair was distributed in the overall local structure of the graph and visualized it in Fig. 5.

The results revealed that the weight of points located at similar angles tends to decrease when applying an angle attention mechanism. This is beneficial in learning local structures by reducing the importance of points in the same direction to block redundant information and increasing the importance of points in different directions.

*D. Ablation Study*

TABLE III
CLASSIFICATION ACCURACY OF ABLATION MODELS ON MODELNET40

| Exp | Ablation model | Accuracy |
|---|---|---|
| 1 | AAG-Net(ours) | 87.24 |
| 2 | Remove Angle MLP | 87.01 |
| 3 | Remove Angle | 84.06 |
| 4 | Remove GL-GPA | 83.21 |

To demonstrate the effectiveness of AAG-Net, we performed an experiment in which modules were removed one by one. Table III shows the experimental results of the angle attention mechanism and GL-GPA. To evaluate the accuracy of AAG-Net trained on GNNs, we started with AAG-Net, which uses both the angle attention mechanism and GL-GPA, and removed each module to determine its impact. The results of the ablation study showed that the angle attention mechanism and GL-GPA blocks are effective and necessary to improve the sampling of AAG-Net. The angle attention mechanism

TABLE IV
EFFICIENCY COMPARISON BETWEEN ORIGINAL AND LIGHTWEIGHT AAG-NET

| Sample size | Method | Layers | Time(m/s) | Accuracy |
|---|---|---|---|---|
| 32 | SampleNet | - | 8.61 | 80.32 |
| | Ours | 4 | 14.28 | 88.25 |
| | Ours(Light) | 3 | 11.62 | 83.95 |

To evaluate the efficiency-performance tradeoff, we also tested a lightweight version of AAG-Net with fewer GNN layers. As summarized in Table IV, this variant maintained competitive performance with reduced inference time.

IV. CONCLUSION

In this letter, we proposed AAG-Net, which consists of two modules, for 3D point cloud sampling. Our model introduced a module that includes point angles from the central point in the attention calculation, allowing it to understand the positional distribution of points and focus on important points. Additionally, by effectively integrating the global features and local details of the point cloud, we designed a sampling structure based on the abstracted object shape. Extensive experimental results on the ModelNet40 dataset showed that our AAG-Net method achieves better performance than state-of-the-art methods in terms of accuracy.

In practical autonomous driving scenarios, however, the robustness of AAG-Net may be affected by sensor noise, missing data, and occlusions. To address these challenges, future research could explore data augmentation strategies—such as injecting simulated noise, point dropout, or artificial occlusions during training—to enhance resilience. Furthermore, geometric denoising techniques, including Statistical Outlier Removal and bilateral filtering, could be employed as preprocessing steps to improve input quality. Moreover, although our proposed AAG-Net was validated specifically for classification tasks, the angle-based attention

mechanism could also significantly contribute to improving 3D object detection tasks in autonomous driving scenarios. Specifically, recent studies such as LSNet [15] demonstrate the critical importance of selecting representative and informative points through learned sampling methods for enhanced object detection performance. Incorporating our angle-based attention into such learned sampling frameworks could potentially provide superior point selections by effectively leveraging angular relationships among neighboring points.

## REFERENCES

[1] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[2] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *in Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.

[4] J. Jeon, S. Hong, H. Lee, J. Kim and J. Yoo, "PMHA-Net: Positional Multi-Head Attention Network for Point-Cloud Part Segmentation and Classification," *in IEEE Access*, vol. 11, pp. 117920-117934.

[5] J. Liu, X. Lv, X. Gong, Y. Liang and J. Hyyppä, "An Unsupervised Learning Network for Large-scale LiDAR Point Clouds Registration," *in IEEE Transactions on Vehicular Technology.*

[6] L. Wang et al., "Multi-Modal and Multi-Scale Fusion 3D Object Detection of 4D Radar and LiDAR for Autonomous Driving," *in IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 5628-5641.

[7] M. Zecchin, M. B. Mashhadi, M. Jankowski, D. Gündüz, M. Kountouris and D. Gesbert, "LIDAR and Position-Aided mmWave Beam Selection With Non-Local CNNs and Curriculum Training," *in IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2979-2990.

[8] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2760–2769.

[9] I. Lang, A. Manor, and S. Avidan, "Samplenet: Differentiable point cloud sampling," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7578–7588.

[10] Y. Ye, X. Yang, and S. Ji, "APSNet: Attention based point cloud sampling," *in Proceedings of the Conference on British Machine Vision Conference (BMVC)*, 2022.

[11] H. Lee, J. Jeon, S. Hong, J. Kim, and J. Yoo, "TransNet: Transformer based point cloud sampling network," *Sensors*, vol. 23, no. 10, p. 4675, May 2023.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *in Proc. 6th Int. Conf. Learn. Representations, 2018.*

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[14] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Wang, M.; Chen, Q.; Fu, Z. LSNet: Learned Sampling Network for 3D Object Detection from Point Clouds. *Remote Sens.* **2022**, *14*, 1539. https://doi.org/10.3390/rs14071539