2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Pyramid Learnable Tokens for 3D LiDAR Place Recognition

Congcong Wen[1], Hao Huang[1] and Yu-Shen Liu[2] and Yi Fang[1]

***Abstract*— 3D LiDAR place recognition plays a vital role in various robot applications,, including robotic navigation, autonomous driving, and simultaneous localization and mapping. However, most previous studies evaluated their models on accumulated 2D scans instead of real-world 3D LiDAR scans with a larger number of points, which limits the application in real scenarios. To address this limitation, we propose a point transformer network with pyramid learnable tokens (PTNet-PLT) to learn global descriptors for an actual scanned 3D LiDAR place recognition. Specifically, we first present a novel shifted cube attention module that consists of a self-attention module for local feature extraction and a cross-attention module for regional feature aggregation. The self-attention module constrains attention computation on a locally partitioned cube and builds connections across cubes based on the shifted cube scheme. In addition, the cross-attention module introduces several learnable tokens to separately aggregate features of points with similar features but spatially distant into an arbitrarily shaped region, which enables the model to capture long-term dependencies of the points. Next, we build a pyramid architecture network to learn multi-scale features and involve a decreasing number of tokens at each layer to aggregate features over a larger region. Finally, we obtain the global descriptor by concatenating learned region tokens of all layers. Experiments on three datasets, including USyd Campus, Oxford Robot-Car, and KITTI, demonstrate the effectiveness and generalization of the proposed model for large-scale 3D LiDAR place recognition.**

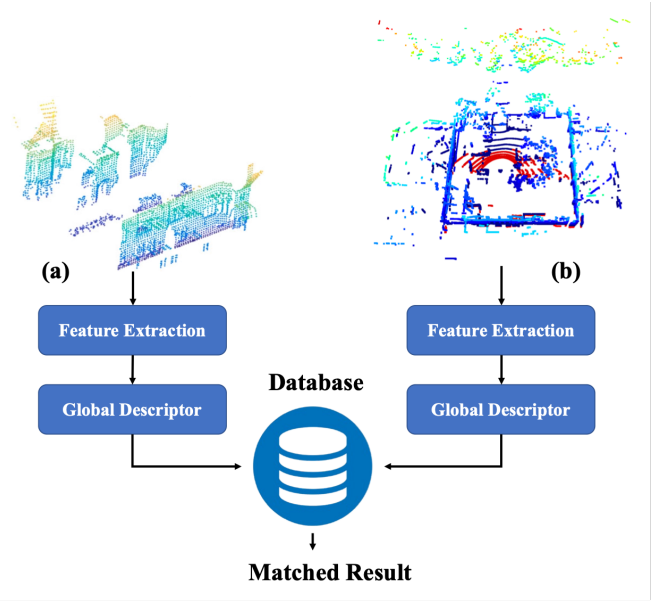

Fig. 1. An illustration of the difference of evaluated scans. (a) an accumulated scan with 4096 equally distributed points that are most commonly adopted by existing studies evaluation; (b) an actual 3D scan with 20k+ not equally distributed points that are additionally evaluated in this paper.

I. INTRODUCTION

Place recognition refers to the ability of an agent to recognize the same place from a different perspective or appearance. This task holds great significance in real-world applications, such as robotic navigation, augmented reality, simultaneous localization and mapping (SLAM), and self-driving vehicles, making it one of the most important tasks in the fields of robotics and computer vision. The core problem of place recognition is to find the robust and effective representation. Among various data sources, 3D LiDAR data has emerged as a popular choice due to its insensitivity to weather and light changes, which enables the learning of a relatively robust representation. Therefore, most of studies have focused on 3D LiDAR-based place recognition.

Early LiDAR-based place recognition methods utilize hand-crafted descriptors to match a query scan with a database of previously acquired template scans. However, the performance of these methods are limited and influenced by the selection of hand-crafted features. Recently, with the significant success of deep learning, an increasing number of researchers shift their attention to using deep neural networks to extract global representation for place recognition.

Due to the irregularity and unorderedness of 3D LiDAR point cloud, previous deep neural networks designed for regular input cannot be directly applied to raw point clouds. Hence, a number of studies first project 3D point cloud into regular formats, such as 2D images [1], [2], [3], 3D voxels [4], [5], [6] and sparse voxelized representations [7], [8], [9], and then leverage existing or modified neural networks to learn global descriptors.

Several deep learning models have been proposed recently, such as PointNet [10], PointNet++ [11] and PointCNN [12], which can be directly applied to raw 3D point clouds. Inspired by this, some studies [13], [14], [15], [16] achieve 3D LiDAR place recognition based on point cloud deep neural networks. For example, PointNetVLAD [13] extracts features using PointNet and aggregates local features into global descriptors using NetVLAD [14] for place recognition. More recently, several works [17], [18], [19] have investigated the application of attention mechanisms in 3D LiDAR place recognition. For instance, PPT-Net [19] leverages a pyramid point transformer module and a pyramid VLAD module

[1]Congcong Wen, Hao Huang and Yi Fang are with NYUAD Center for Artificial Intelligence and Robotics, New York University Abu Dhabi, UAE, and NYU Multimedia and Visual Computing Lab, New York University, USA, and New York University Abu Dhabi, UAE. cw3437@nyu.edu, hh1845@nyu.edu and yf23@nyu.edu

[2]Yu-Shen Liu is with the School of Software, Tsinghua University, Beijing, P. R. China. liuyushen@tsinghua.edu.cn

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161523

to respectively learn local features and aggregate context information into the global descriptors.

However, most of the above methods are evaluated on accumulated 2D scans consisting of 4096 points from the Oxford RobotCar dataset, which is not compatible with scans from a 3D LiDAR commonly used in real-world applications. In order to improve the practicality of the methods in real scenarios, we choose the same dataset as MinkLoc3D-SI [9] to train and evaluate our model. Different from the MinkLoc3D-SI that projects 3D point cloud into a sparse voxelized representation, our model directly inputs raw point cloud and utilizes a novel shifted cube attention module to efficiently learn local features and capture long-term dependencies. Moreover, we introduce multi-stage learnable tokens to capture regional features at different resolutions, improving the quality of learned global descriptor.

In this paper, we propose a point transformer network with pyramid learnable tokens (PTNet-PLT) to learn global descriptor for 3D LiDAR place recognition. Specifically, we first employ a point embedding layer to aggregate the geometric and contextual information for each point from its local neighbors. Then, we build a pyramid architecture network to learn multi-scale features. Within each scale, we present a novel shift cube attention module that consists of self-attention module to efficiently extract local features and cross-attention module to aggregate regional features into learnable tokens; and across different scales, we involve a decreasing number of tokens to capture regional features at different resolutions. Finally, we obtain the global descriptor by concatenating learned region tokens of all layers. The main contributions of this paper are summarized as follows:

- We develop a point transformer network to learn global descriptors from raw point clouds for place recognition, which is more suitable for actual scans from 3D LiDAR.
- We introduce a novel shifted cube attention module to efficiently learn local features and aggregate features of points with similar features but spatially distant into an arbitrarily shaped region, which enables model to capture long-term dependencies of the points.
- We involve a decreasing number of tokens at each layer to capture regional features at different resolutions, and concatenate learned region tokens of all layers to the global descriptor.
- Experiments demonstrate the proposed model achieves state-of-the-art performance for place recognition on three datasets, including USyd Campus, Oxford RobotCar and KITTI.

II. Related Work

A. 3D LiDAR Representation Learning

Early works mainly apply traditional machine learning algorithms, including Support Vector Machine, Random Forest and Conditional Random Field, to obtain 3D LiDAR point cloud representation based on hand-crafted geometric features, such as curvature, normal, roughness and point feature histograms. However, these methods require manual feature calculation and are sensitive to different features.

Recently, researchers have turned to deep learning models to benefit from their automatic feature extraction capabilities. Deep learning models for 3D LiDAR representation learning can be categorized into two categories: Image/Voxel-based and Point Cloud-based methods.

Image/Voxel based methods. Due to the irregular and unordered nature of 3D LiDAR point cloud, some studies first transform LiDAR point cloud to regular images or voxels. For example, MV3D [2] first generates the Bird's Eye View (BEV) images and Front View (FV) images from LIDAR point cloud, and then adopts image-based neural network to acquire fusion features. Similarly, [1] and [3] convert point clouds into front-view 2D maps and 2D Bird's Eye View (BEV) images respectively, and utilize image-based detectors to learn features. Although these methods achieve satisfactory performance, projecting to a specific viewpoint can lead to information loss, particularly in complex scenes. Instead of generating images, various works, including Vote3deep [4], PointPillars [5] and SECOND [6], convert 3D LiDAR point cloud into 3D voxels and employ existing convolutional neural network models to learn representation. Nonetheless, choosing a proper voxel resolution is important since higher resolutions may result in more information loss, whereas lower resolutions may pose computational and memory challenges.

Point Cloud based methods. Recently, PointNet model [10], a pioneer study that directly applies to 3D point clouds, adopts the Multi-Layer Perceptron (MLP) and the symmetric function to encode global representation. Following this work, a number of PointNet-like architecture networks [11], [12], [20], [21] have been proposed. Therefore, point cloud based deep neural networks have received increased attention to achieve 3D LiDAR representation learning. D-FCN [22] designs a fully convolutional neural network under directionally constrained to extract multi-scale features from original LiDAR point cloud. Besides, GACNN [23] presents a global-local graph attention convolution neural network that is directly applied to 3D LiDAR point cloud to conduct classification based on learned representation. In addition, PointRCNN [24] proposes a two-stage detection framework for detecting 3D objects from irregular LiDAR point cloud.

B. 3D LiDAR Place Recognition

Consistent with the 3D LiDAR representation learning approaches, existing algorithms for 3D LiDAR place recognition can also be divided into Image/Voxel based methods and Point Cloud based methods.

Image/Voxel based methods. [25] generates Scan Context Image (SCI) from LiDAR point cloud and presents a CNN-based end-to-end localization framework. And BV-Match [26] projects 3D Lidar scans to bird's-eye view (BEV) images and introduces a novel descriptor, Bird's-eye View Feature Transformer (BVFT), which is built based on Log-Gabor filters and maximum index map (MIM). In addition, VBRL [27] divides the input 3D point cloud into voxels and extracts multi-modal features from these voxels to con-

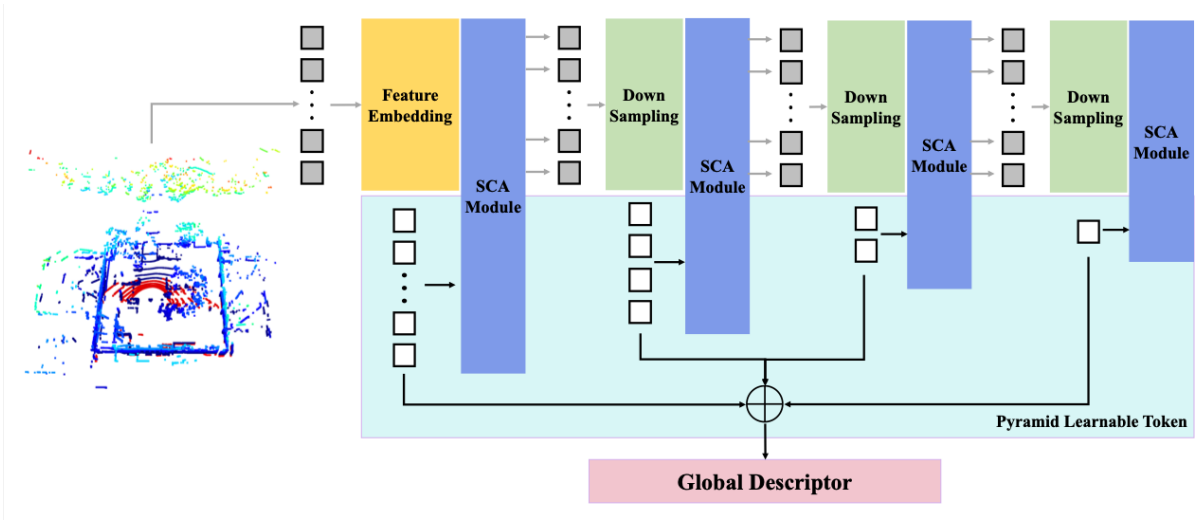

Fig. 2. Illustration of the overall architecture of the PTNet-PLT model. The point cloud is first input to a Feature Embedding Module to aggregate local geometric information. And then the updated features are passed into the proposed Shifted Cube Attention (SCA) Module for feature extraction. Next, the learned features go through three layers, each of which consists of a Down Sampling Module and a Shifted Cube Attention (SCA) Module. Meanwhile, the learnable tokens of each stage aggregate local features to arbitrarily shaped region features for capturing the long-term dependencies of point clouds. Finally, the learned region descriptor of learnable token from different stages are concatenated into a final global descriptor.

duct place recognition. Moreover, MinkLoc3D [7] quantizes LiDAR point cloud into a sparse voxelized representation and adopts sparse convolutions to extract global features efficiently. Similarly, MinkLoc++ [8] fuses camera images and the sparse voxelized representation of 3D LiDAR scans to further improves MinkLoc3D by involving a channel attention mechanism. Recently, MinkLoc3D-SI [9] takes the spherical representation of 3D LiDAR point cloud and intensity value as input, and adopt sparse convolution neural network to acquire global descriptors.

Point Cloud based methods. PointNetVLAD [13] aggregates local features into global descriptors for place recognition based on PointNet and NetVLAD [14]. In order to enable the extracted features are related to corresponding task, PCAN [15] proposes a context-aware reweighting network to learn global descriptor for 3D point cloud. Besides, LPD-Net [16] utilizes a graph-based neural network to adaptively learn local features based on their spatial distribution.

More recently, various studies leverage self-attention mechanism to capture global features of LiDAR point cloud for place recognition. [17] designs a Dual Attention module to distinguish local features that are relevant to different tasks. These local features are then further aggregated by a Residual Graph Convolution Network module to obtain final representation. [18] introduces a NDT-Transformer network to learn global descriptors from a set of 3D Normal Distribution Transform (NDT) cell representations generated from raw 3D point cloud. [28] presents a point orientation encoding module to learn local features effectively by taking into account the relationship each point and its neighbors, and further introduces a self-attention unit that distinguishes the relative importance of different local features to the global descriptors. PPT-Net [19] develops a pyramid point transformer module to adaptively learn the spatial relationship of neighboring points of each point and constructs a pyramid VLAD module to aggregate the multi-scale context information into the global descriptors.

III. Methods

A. Overview

In this section, we build a point transformer network with pyramid learnable tokens (PTNet-PLT) to learn global descriptor for 3D LiDAR place recognition, the overall architecture of which is illustrated in Fig 2. Given the non-uniform density of the scanned point cloud P, each point in the cloud is composed of raw coordinates (x, y, z), spherical coordinates (r, θ, ϕ), and intensity (I), as expressed by the following formulation:

$$p_i = [x, y, z, r, \theta, \phi, I], \quad p_i \in P, \tag{1}$$

where r represents the range between the scanner and scanned point, θ indicates the angle of horizontal scanning, and ϕ denotes the angle of vertical scanning.

First, we employ the *Feature Embedding Module* to aggregate the geometric and local information from neighbors of each point. Then, we feed the updated features and various learnable tokens to the *Shifted Cube Attention Module* to further aggregate local geometric features and contextual information to several region features. Next, we leverage three layers, each of which consists of: a *Down Sampling Module* and a Shifted Cube Attention Module, to learn multi-scale features of the point cloud. Meanwhile, we introduce *Pyramid Learnable Token* to capture regional features at different resolutions. Finally, we concatenate the learnable tokens from different stages into a global descriptor.

B. Feature Extraction Modules

1) Feature Embedding Module: Before feeding the raw point cloud into neural network, most of studies employ

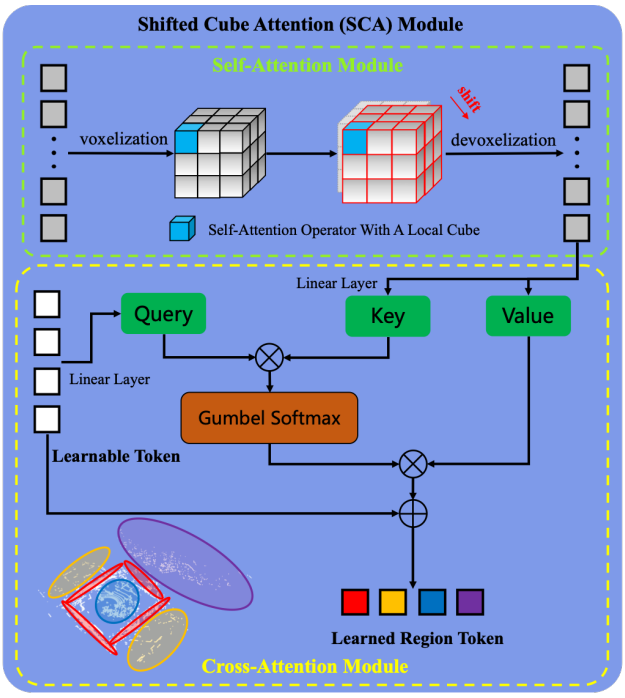

Fig. 3. An illustration of the Shifted Cube Attention (SCA) Module that consists of Self-Attention Module and Cross-Attention Module. The region where the features learned by each learnable token is circled by a corresponding colored ellipse at the bottom-left corner.

an additional layer to project the input features to a high dimension. And [29] shows that using a linear layer or MLP will result in slow convergence and poor performance. Therefore, we utilize KPConv [30] to aggregate local geometric information from neighbors of each point. The outputs are passed to the following modules to extract deeper features.

2) Shifted Cube Attention Module: To enhance the quality of descriptors used for place recognition, the shifted cube attention (SCA) module incorporates a self-attention module to capture local features, and a cross-attention module to encode local point features into arbitrarily shaped region features. The architecture of SCA module is shown in Fig. 3.

Self-Attention Module: As described before, our goal is to achieve place recognition on an actual scan of 3D LiDAR with a large number of points, which will bring a great challenge for computation. To remedy this issue, we partition point cloud into non-overlapping cubes and conduct multi-head self-attention within a local cube to learn local descriptors. Moreover, inspired by Swin Transformer[31], we further build connections across cubes based on shifted cube scheme. Compared to the complex shifted window operation for 2D images, shifted cube operation for 3D point cloud can be easily achieved by shifting the point cloud by half of the cube size and re-partitioning shifted point cloud into new non-overlapping cubes, which is illustrated in the green dashed box at the top of Fig. 3. Finally, we propagate the learned features to each point.

Cross-Attention Module: After getting the learned local point features, we introduce a number of learnable tokens to capture long-term dependencies of the points by aggregating individual point features to arbitrarily shaped region features. Note that the region represented by each token can be discontinuous, which ensures that points of distant but similar categories are aggregated together. Specifically, we first use linear layer to project the features of learnable token to query and learned point features to key and value. Then we calculate the similarity matrix between all points and learnable tokens, and leverage gumbel softmax to assign each point to corresponding token. Next we merge the features of all points belonging to the same token to generate new features that represents a discontinuous region. Finally, we obtain the region features by adding a residual connection.

3) Down Sampling Module: Given an input point cloud, we first generate a subset of points with its features based on the raw coordinates by adopting the farthest point sampling (FPS) algorithm. And then we group the features for each down-sampled point from its neighboring points, which is selected by using K nearest neighbor (KNN) algorithm. Finally, we employ max pooling operator to aggregate the grouped features of neighbor points to obtain the representation of down-sampling point cloud.

4) Pyramid Learnable Token: As previously introduced, we present several learnable tokens to aggregate local point level features into arbitrarily shaped region level features at one SCA module. Meanwhile, to capture multi-scale features, we adopt three down sampling layers with each layer followed by a SCA module to obtain a latent representation at the current scale. As a result, the learnable tokens of different levels are able to learn regional features with different resolutions. Considering that the number of point clouds becomes sparser after down sampling layer, we also gradually reduce the number of tokens in each layer. After the last down sampling layer, we use only one learnable token to aggregate the features of the whole point cloud. Finally, we project the features of the learnable tokens in each layer to the same dimension and concatenate them to obtain the final global descriptor.

C. Loss Function

Our neural network is trained using triplet margin loss function. The training objective is to minimize the distance between the anchor point cloud descriptor a_i and its corresponding positive descriptor p_i, while maximizing the distance between a_i and negative descriptor n_i. The loss function can be defined as follows:

$$L = \sum_i \max(D(a_i, p_i) - D(a_i, n_i) + M, 0), \qquad (2)$$

where $D(\cdot, \cdot)$ denotes Euclidean distances and M represents hyperparameter margin that controls the degree of separation between the positive and negative samples.

IV. Experiments

A. Datasets

1) USyd Campus: The USyd Campus Dataset [32] is collected from a buggy-like car with Velodyne VLP-16 LiDAR at the University of Sydney campus and surroundings.

It contains more than 60 weeks drives in varying weather conditions. [9] processes the raw dataset and generates a new dataset for place recognition. In new dataset, each location of 3D LiDAR scans contains up to 25,000 points with raw 3D coordinates and intensity values. Following the setting of [9], we use 19,138 point clouds for training and 8,797 point clouds for testing.

2) Oxford RobotCar Intensity: The Oxford RobotCar Intensity dataset [9] is built from the Oxford RobotCar dataset [33] that consists of 1010.46 km and over a year of recorded driving in central Oxford, UK. [13] generates a dataset for 3D place recognition from 2D accumulated scans based on 44 sets of full and partial runs of the Oxford RobotCar dataset. And [9] modifies this dataset by adding intensity information. Similar to these two works, we employ 21,711 point clouds to train our network and 3,030 point clouds to test.

3) KITTI: In order to evaluate the generalization ability of the proposed model, we directly utilize our network that is trained on other datasets, such as USyd Campus dataset and Oxford RobotCar Intensity dataset. According to [8], we build the reference database by revisiting the same places appeared in the Sequence 00's first 170 seconds, and the remainder of the Sequence 00 is employed as queries.

B. Evaluation metrics

We adopt Average Recall@N (AR@N) as the evaluation metric to measure the accuracy of place recognition algorithms. And we assume that the location is succeeded to be recognized if one or more places from the first N point clouds retrieved are less than a certain distance threshold from the query point cloud. For the USyd dataset, this threshold is set to 10m, while for the Oxford RobotCar and KITTI datasets, it is set to 25m. We present the results of our experiments in terms of both AR@1% and AR@1.

C. Results

1) USyd Campus: According to the above descriptions, the distance threshold for determining whether the location is correctly identified is set to 10m on the USyd Campus dataset, which is stricter than 25m that is commonly used in other two datasets. We report the performance of the proposed model and recently state-of-the-art methods, including MinkLoc3D, Scan Context, MinkLoc3D-I, MinkLoc3D-S and MinkLoc3D-SI on USyd Campus dataset for place recognition in Table I. It can be found that the proposed model has achieved the best performance on both metrics of $AR@1\%$ and $AR@1$.

2) Oxford RobotCar Intensity: We first evaluate the performance of our methods and various versions of MinkLoc3D on the modified Oxford RobotCar Intensity dataset in Table II. It can be found that MinkLoc3D-S with spherical coordinates is less effective than MinkLoc3D with Cartesian coordinates on this dataset. On the contrary, our model takes both coordinates as input and automatically selects the more suitable coordinate expression during feature learing, thus achieving the best recognition performance.

TABLE I

QUANTITATIVE COMPARISONS BETWEEN THE PROPOSED MODEL AND OTHER STATE-OF-THE-ART MODELS ON THE USYD DATASET.

USyd dataset	$AR@1\%$	$AR@1$
MinkLoc3D [7]	98.1	91.7
Scan Context [34]	88.7	86.0
MinkLoc3D-I [9]	98.2	92.3
MinkLoc3D-S [9]	98.8	93.9
MinkLoc3D-SI [9]	99.0	94.7
PTNet-PLT	**99.4**	**96.4**

Moreover, in order to compare fairly with more previous methods, we also test our method on the original Oxford RobotCar dataset, and the results are listed in Table III. It can be found that our model still achieves the best recognition accuracy compared to the existing models.

TABLE II

QUANTITATIVE COMPARISONS BETWEEN THE PROPOSED MODEL AND OTHER STATE-OF-THE-ART MODELS ON THE OXFORD ROBOTCAR INTENSITY DATASET.

Oxford RobotCar Intensity	$AR@1\%$	$AR@1$
MinkLoc3D [7]	97.6	92.8
MinkLoc3D-I [9]	98.1	93.6
MinkLoc3D-S [9]	92.0	79.9
MinkLoc3D-SI [9]	93.4	82.2
PTNet-PLT	**98.5**	**94.2**

TABLE III

QUANTITATIVE COMPARISONS BETWEEN THE PROPOSED MODEL AND OTHER STATE-OF-THE-ART MODELS ON THE ORIGINAL OXFORD ROBOTCAR DATASET.

Oxford RobotCar	$AR@1\%$	$AR@1$
PointNetVLAD [13]	80.3	63.3
Scan Context [34]	81.9	64.6
PCAN [15]	83.8	70.7
DH3D-4096 [35]	84.3	73.3
EPC-Net [36]	94.7	86.2
LPD-Net [16]	94.9	86.4
DISCO [37]	75.0	88.4
SOE-Net [28]	96.4	89.3
NDT-Transformer [18]	97.7	93.8
TransLoc3D [38]	98.5	95.0
MinkLoc3D (3D) [7]	97.9	93.8
MinkLoc3D-S [9]	93.1	82.0
PTNet-PLT	**98.9**	**95.6**

3) KITTI: Similar to the previous work [13], [16], [8], [9], to demonstrate the generalization capability of the proposed method, we train our model on the Oxford RobotCar Intensity dataset or USyd Campus dataset and then test on the KITTI dataset. We list the results of our model and existing models on KITTI dataset in Table IV. As can be seen from the Table IV, our model still achieves the highest accuracy for place recognition on KITTI dataset, regardless of whether the training data is from Oxford RobotCar Intensity dataset or USyd Campus dataset. Therefore, it can be seen that our model achieves a satisfactory generalization capability.

TABLE IV

QUANTITATIVE COMPARISONS OF GENERALIZATION RESULTS BETWEEN THE PROPOSED MODEL AND OTHER STATE-OF-THE-ART MODELS ON THE KITTI DATASET.

KITTI dataset	Trained	$AR@1\%$	$AR@1$
PointNetVLAD [13]	Oxford RC	72.4	–
LPD-Net [16]	Oxford RC	74.6	–
MinkLoc++ (3D) [8]	Oxford RC	72.6	–
Scan Context [34]	-	75.0	71.4
MinkLoc3D [7]	USyd	73.8	69.1
MinkLoc3D-SI [9]	Oxford RCI	81.0	72.6
MinkLoc3D-SI [9]	USyd	81.0	78.6
PTNet-PLT	Oxford RCI	83.4	74.6
PTNet-PLT	USyd	**85.0**	**82.6**

D. Ablation study

1) Effect of Feature Embedding Module and Shifted Cube Attention Module.: In this section, we design an ablation experiment to evaluate the effectiveness of feature embedding module (FEM) and shifted cube attention module (SCA). To further explore the validity of shifted cube attention module, we discuss the case where it contains a shifted cube operation. We show the results of the proposed module in four different cases on the USyd Campus dataset in Table V. By comparing the first case with the third case, the second case with the fourth case, we can find that feature embedding module (FEM) can slightly improve the accuracy (around 0.7% on $AR@1\%$). Similarly, by comparing the first case with the second case, the third case with the fourth case, we can find that the shifted cube operation of shifted cube attention module (SCA) can greatly improve the performance for place recognition.

TABLE V

THE RESULTS OF PTNET-PLT WITH DIFFERENT MODULES ON THE USYD CAMPUS DATASET.

FEM	SCA (w/o shift)	SCA (w/ shift)	$AR@1\%$	$AR@1$
✗	✓	✗	97.1	93.0
✗	✗	✓	98.8	95.5
✓	✓	✗	97.9	94.5
✓	✗	✓	**99.4**	**96.4**

2) Effect of Pyramid Learnable Token.: We discuss the effectiveness of pyramid learnable token in this section. As introduced before, in order to capture long-term dependencies of points within a given point cloud, we involve a number of tokens at each layer to aggregate points with similar features into an arbitrarily shaped region that can be disconnected. As the number of layers increases, we progressively reduce the number of tokens per layer so that each token can represent a larger region. To verify the effectiveness of this architecture design, we conduct an ablation study by comparing with other two architectures, i.e. global feature extraction with pooling layers and global feature extraction with a fixed number of tokens, which are illustrated in Fig. 4. We compare the above two architectures with pyramid learnable token in Table VI. From Table VI, we can find that the architecture of pyramid learnable token achieves better performance than other two architectures. Moreover, it can be also found that the number of tokens should not be too large or too small if a fixed number of tokens is used to extract features.

TABLE VI

THE RESULTS OF PTNET-PLT WITH DIFFERENT ARCHITECTURES ON THE USYD CAMPUS DATASET.

Architecture	$AR@1\%$	$AR@1$
Pooling Layer	92.3	91.1
N Learnable Token (N=1)	95.8	92.5
N Learnable Token (N=2)	96.9	94.6
N Learnable Token (N=4)	96.3	96.7
N Learnable Token (N=8)	95.9	93.5
Pyramid Learnable Token	**99.4**	**96.4**

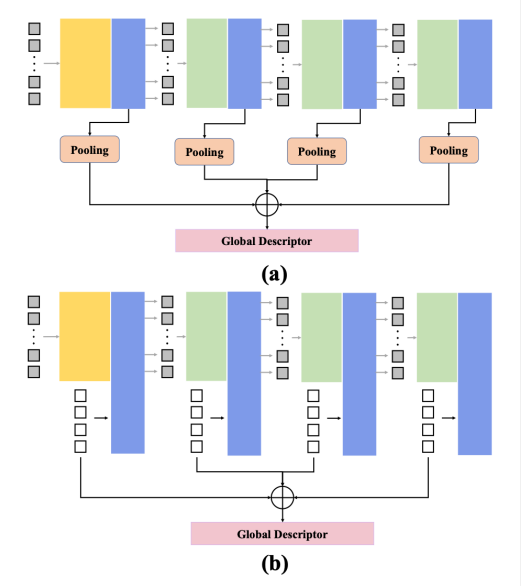

Fig. 4. An illustration of the architecture of two baseline methods. (a) using pooling layer at each stage to extract global descriptor; (b) using a fixed number of tokens at each stage to extract global descriptor.

V. CONCLUSIONS

In this paper, we propose a point transformer network with pyramid learnable tokens (PTNet-PLT) to learn global descriptors for 3D LiDAR place recognition. In order to improve the practicality of the proposed model in the real world, we select actual 3D scans containing a larger number of points as the evaluated dataset. To efficiently obtain the local features, we introduce learnable tokens and propose a novel shifted cube attention module that consists of self-attention module for local features extraction and cross-attention module for region features aggregation and long-term dependencies capture. Moreover, to acquire regional features at different resolutions, we involve learnable tokens at each layer and gradually reduce the number of tokens. The final global descriptor for place recognition is obtained by concatenating all region tokens of each layer. Experiments on three datasets, including USyd Campus, Oxford Robot Car and KITTI, demonstrate the effectiveness and generalization of the proposed model for 3D LiDAR place recognition.

REFERENCES

[1] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv preprint arXiv:1608.07916*, 2016.

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[3] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: a 3d object detection framework from lidar information," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3517–3523.

[4] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1355–1361.

[5] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[6] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[7] J. Komorowski, "Minkloc3d: Point cloud based large-scale place recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1790–1799.

[8] J. Komorowski, M. Wysoczańska, and T. Trzcinski, "Minkloc++: lidar and monocular image fusion for place recognition," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[9] K. Żywanowski, A. Banaszczyk, M. R. Nowicki, and J. Komorowski, "Minkloc3d-si: 3d lidar place recognition with sparse convolutions, spherical coordinates, and intensity," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1079–1086, 2021.

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

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[12] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, pp. 820–830, 2018.

[13] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4470–4479.

[14] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[15] W. Zhang and C. Xiao, "Pcan: 3d attention map learning using contextual information for point cloud based retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 436–12 445.

[16] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y.-H. Liu, "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2831–2840.

[17] Q. Sun, H. Liu, J. He, Z. Fan, and X. Du, "Dagc: Employing dual attention and graph convolution for point cloud based place recognition," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, pp. 224–232.

[18] Z. Zhou, C. Zhao, D. Adolfsson, S. Su, Y. Gao, T. Duckett, and L. Sun, "Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5654–5660.

[19] L. Hui, H. Yang, M. Cheng, J. Xie, and J. Yang, "Pyramid point cloud transformer for large-scale place recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6098–6107.

[20] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[21] X. Li, C. Wen, L. Wang, and Y. Fang, "Topology constrained shape correspondence," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 10, pp. 3926–3937, 2020.

[22] C. Wen, L. Yang, X. Li, L. Peng, and T. Chi, "Directionally constrained fully convolutional neural network for airborne lidar point cloud classification," *ISPRS journal of photogrammetry and remote sensing*, vol. 162, pp. 50–62, 2020.

[23] C. Wen, X. Li, X. Yao, L. Peng, and T. Chi, "Airborne lidar point cloud classification with global-local graph attention convolution neural network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 181–194, 2021.

[24] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[25] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[26] L. Luo, S.-Y. Cao, B. Han, H.-L. Shen, and J. Li, "Bvmatch: Lidar-based place recognition using bird's-eye view images," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6076–6083, 2021.

[27] S. Siva, Z. Nahman, and H. Zhang, "Voxel-based representation learning for place recognition based on 3d point clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8351–8357.

[28] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla, "Soe-net: A self-attention and orientation encoding network for point cloud based place recognition," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2021, pp. 11 348–11 357.

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

[30] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[31] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[32] W. Zhou, J. S. Berrio Perez, C. De Alvis, M. Shan, S. Worrall, J. Ward, and E. Nebot, "The usyd campus dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/sk74-7419

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[34] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[35] J. Du, R. Wang, and D. Cremers, "Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 744–762.

[36] L. Hui, M. Cheng, J. Xie, J. Yang, and M.-M. Cheng, "Efficient 3d point cloud feature learning for large-scale place recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 1258–1270, 2022.

[37] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, "Disco: Differentiable scan context with orientation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2791–2798, 2021.

[38] T.-X. Xu, Y.-C. Guo, Y.-K. Lai, and S.-H. Zhang, "Transloc3d: Point cloud based large-scale place recognition using adaptive receptive fields," *arXiv preprint arXiv:2105.11605*, 2021.