Pyramid Point Cloud Transformer for Large-Scale Place Recognition

Le Hui, Hang Yang, Mingmei Cheng, Jin Xie* and Jian Yang*
PCA Lab, Nanjing University of Science and Technology, China
{le.hui, hangyang, chengmm, csjxie, csjyang}@njust.edu.cn

Abstract

Recently, deep learning based point cloud descriptors have achieved impressive results in the place recognition task. Nonetheless, due to the sparsity of point clouds, how to extract discriminative local features of point clouds to efficiently form a global descriptor is still a challenging problem. In this paper, we propose a pyramid point cloud transformer network (PPT-Net) to learn the discriminative global descriptors from point clouds for efficient retrieval. Specifically, we first develop a pyramid point transformer module that adaptively learns the spatial relationship of the different k-NN neighboring points of point clouds, where the grouped self-attention is proposed to extract discriminative local features of the point clouds. The grouped self-attention not only enhances long-term dependencies of the point clouds, but also reduces the computational cost. In order to obtain discriminative global descriptors, we construct a pyramid VLAD module to aggregate the multi-scale feature maps of point clouds into the global descriptors. By applying VLAD pooling on multi-scale feature maps, we utilize the context gating mechanism on the multiple global descriptors to adaptively weight the multi-scale global context information into the final global descriptor. Experimental results on the Oxford dataset and three in-house datasets show that our method achieves the state-of-the-art on the point cloud based place recognition task. Code is available at `https://github.com/fpthink/PPT-Net`.

1. Introduction

Place recognition is an important task in the computer vision and robotics communities, and has been widely applied to many fields such as autonomous driving [19, 20, 26], augmented reality [34], robot navigation [16, 33, 27], and simultaneous localization and mapping (SLAM) [2, 14, 29]. Place recognition is mainly divided into two categories: image-based and point cloud-based. For image-based place recognition, given a query image of a local scene, the goal is to retrieve the best match in the database, so that the exact location of the query image relative to the reference map of the scenarios can be determined. However, the image-based place recognition is sensitive to environmental changes, such as seasonal and illumination changes. Therefore, point cloud-based place recognition approaches are proposed to alleviate these limitations by using 3D point clouds.

In the past few years, with the development of various point cloud processing methods [36, 38, 50, 46, 4], many efforts have been made for point cloud based place recognition. Mikaela *et al.* [47] proposed PointNetVLAD, which first uses PointNet [36] to extract point features and then adopts NetVLAD [1] to generate global descriptors for retrieval. Based on PointNetVLAD, Zhang *et al.* [57] proposed the point contextual attention network (PCAN) to predict the significance of point features to generate discriminative global descriptors. However, the point features of these methods are obtained through PointNet, which cannot capture the local geometric structures of the point clouds. Liu *et al.* [30] proposed a large-scale location description network (LPD-Net), which employs a graph-based aggregation module in both coordinate and feature spaces to extract the local features of the point clouds by combining the handcrafted features of the point clouds. Lately, Xia *et al.* [52] proposed a self-attention and orientation encoding network (SOE-Net) that uses the self-attention unit to capture the spatial relationship of the point clouds to enhance local features. Nevertheless, the point-wise self-attention operation cannot fully exploit the neighboring structure of each point to capture the long-term dependencies between different regions of the point clouds well. Particularly, since the multi-scale structure information of the regions is not incorporated in the generated local descriptors, they may not be discriminative enough to characterize the regions in the point clouds with varying densities.

*Corresponding authors

Le Hui, Hang Yang, Mingmei Cheng, Jin Xie and Jian Yang are with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, China.

In this paper, we propose a pyramid point cloud transformer network (PPT-Net) to learn global descriptors from point clouds with increasing contextual scales. Specifically, we first develop the pyramid point transformer module to generate the discriminative local features of the point clouds. In the pyramid point transformer module, we apply EdgeConv [50] on the multi-scale k-nearest neighbor (k-NN) graphs of each point to extract the local embeddings at different scales. Based on the local embedding at each scale, we propose the grouped self-attention to adaptively learn the spatial relationship between different regions. In grouped self-attention, we divide the local embedding into mutually independent groups along the channel dimension through group convolution, and compute the similarity between different regions for each group. Thus, the points with similar neighboring structures will be assigned to the high weights so that the discrimination of the local features of the points can be boosted. In addition, the grouped self-attention can reduce the computational costs through the grouping operation. After generating discriminative local features of the point clouds, we then develop a pyramid VLAD module to aggregate local features into the discriminative global descriptors. In the pyramid VLAD module, we apply the VLAD pooling [1] on multi-scale feature maps to generate multi-scale global descriptors. Based on the multi-scale global descriptors, we apply the context gating mechanism on the multiple global descriptors to adaptively weight the multi-scale global context information into the final global descriptor. For place recognition, the experimental results on the Oxford dataset and three in-house datasets show that our method achieves new state-of-the-art.

The contributions of this paper are as follows:

- We develop a pyramid point transformer module to adaptively learns the spatial relationship between different regions of point clouds at different scales by using the grouped self-attention to extract discriminative local descriptors.
- We develop a pyramid VLAD module to aggregate multi-scale feature maps of point clouds into the discriminative global descriptor.
- The proposed PPT-Net can achieve the state-of-the-art on various benchmark datasets for the point cloud based place recognition task.

2. Related Work

3D local descriptors. How to extract powerful local descriptors is a key problem in many 3D vision tasks, such as 3D object matching and reconstruction. Spin image [22] is a regional descriptor that converts the local neighboring points into 2D spin images to characterize the 3D shapes of 3D objects. Geometry histogram [15] introduces 3D shape contexts and harmonic shape contexts to improve the recognition of 3D objects in noisy and cluttered scenes. Point feature histograms (PFHs) [42] and fast point feature histograms (FPFHs) [41] use a multi-dimensional histogram to encode the k-neighborhood geometrical properties of each point. Signature of histogram of orientation (SHOT) [43] is a 3D local descriptor for surface matching, which encodes histograms of the geometric information of the local neighborhood of the points to obtain local descriptors.

Recently, several efforts of local 3D shape descriptors have been made on multi-view representation (multi-view images) and 3D volumetric representation (voxel), where the 2D/3D convolutional neural networks (CNNs) are directly applied to learn feature embeddings. Based on volumetric representation, Volumetric CNN [37], 3D ShapeNets [51] and OctNet [39] are proposed for 3D object classification. 3DMatch [56] learns a local volumetric patch descriptor for 3D correspondence. In addition to the volumetric representation, multi-view convolutional neural network (MVCNN) [44] projects the point clouds into multi-view images, and then apply 2D CNNs for object recognition. Recently, Qi *et al.* proposed PointNet [36], a pioneering network that makes it possible to take 3D points as the input. Based on PointNet, Deng *et al.* [9] proposed the point pair feature network (PPFNet) to learn the 3D local feature descriptors to find correspondences in the point clouds. Based on folding-based auto-encoder, Deng *et al.* [8] proposed PPF-FoldNet [8] that integrates a PointNet encoder with a FoldingNet decoder to learn rotation invariant 3D local features without supervision. Based on PPF-FoldNet, Deng *et al.* [10] proposed a relative pose estimation network (RelativeNet) to assign correspondence-specific orientations to the key points to generate pose-related descriptors. Yew *et al.*[55] proposed the 3DFeat-Net that uses alignment and attention mechanisms to learn 3D feature descriptors from the GPS tagged 3D point clouds. Based on the smoothed density value (SDV) voxelization representation, 3DSmoothNet [17] learns a compact local feature descriptor for 3D point cloud matching.

3D global descriptors. In the place recognition task, 3D global descriptors are usually used to characterize the entire scene. Handcrafted global descriptors usually use statistical information of the LiDAR data to describe the scenes. Rohling *et al.* [40] used the static histograms extracted from the 3D LiDAR points to recognize 3D scenes. Cop *et al.* [6] proposed the DELIGHT, a descriptor of LiDAR intensities as a group of histograms, to encodes intensity information of the LiDAR data into histograms to obtain the global descriptor. Cao *et al.* [3] converted 3D laser points to bearing angle images and aggregate the oriented fast and rotated brief (ORB) features extracted from images to obtain a global descriptor.

Recently, many efforts [13, 45, 21] have been made

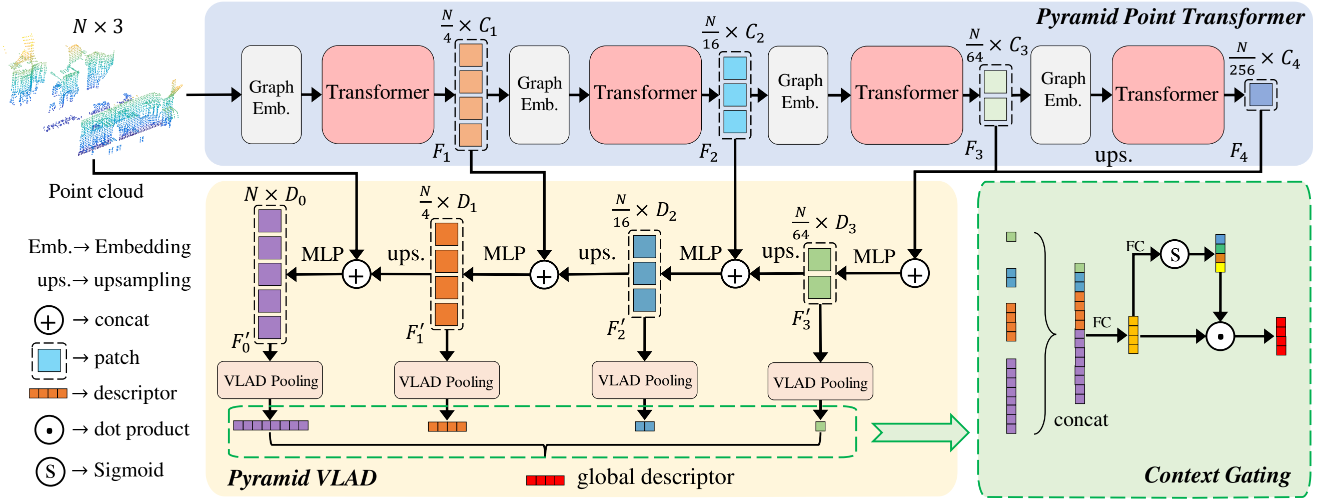


Figure 1: The pipeline of the pyramid point cloud transformer network (PPT-Net). Given the point cloud, we first utilize the pyramid point transformer to capture the spatial relationship of the point clouds at different resolutions to enhance the discrimination of local features of the point clouds. Then, we construct a pyramid VLAD module to aggregate the multi-scale descriptors generated by VLAD pooling into a discriminative global descriptor through context gating for retrieval.

on learning global descriptors for point cloud based place recognition. Inspired by PointNet [36] and NetVLAD [1], Mikaela *et al.* [47] proposed PointNetVLAD to learn global descriptors of point clouds for retrieval. Subsequently, point contextual attention network (PCAN) [57] improves PointNetVLAD by learning an attention map to generate global descriptors. Since both of them use PointNet to extract point features, they cannot capture the local geometric structures of point clouds well. Therefore, Liu *et al.* [30] proposed the large-scale place description network (LPD-Net) to capture the local geometric structures of the point clouds by using a graph-based aggregation module in both coordinate and feature spaces. Like PointNetVLAD, LPD-Net uses the NetVLAD layer to obtain global descriptors of the point clouds for place recognition. MinkLoc3D [23] uses sparse 3D convolutional neural networks (3D CNNs) on sparse voxelized point clouds to extract local features of point clouds. For place recognition, it uses a simple generalized-mean pooling [24] layer to aggregate the local features into a global descriptor. Lately, Xia *et al.* [52] proposed a self-attention and orientation encoding network (SOE-Net) to capture the spatial relationship of point clouds through self-attention unit. It also uses the NetVLAD layer to extract the global descriptors for retrieval.

Transformer. Transformer family [48, 11, 7, 54] has been widely used in neural machine translation. As a pioneer, transformer [48] uses the self-attention mechanism [25] to capture long-term dependencies of language sequence without using recurrence or convolution operations. Subsequently, Devlin *et al.* [11] proposed the bidirectional encoder representation from transformers (BERT), which considers both the left and right context of sequences in all layers in the transformer. Recently, transformer [28, 49, 53] is extended to the 2D vision. Vision transformer (ViT) [12] first divides an image into patches and then feeds the sequence of linear embeddings of these patches to a transformer for image classification. Point transformer networks [18, 58] have achieved good results in point cloud semantic segmentation. Nonetheless, due to the high computational complexity of self-attention, it is difficult to use these networks to tackle large-scale point clouds.

3. Method

3.1. Pyramid Point Transformer

Overall architecture. For place recognition, the pyramid point transformer module aims to capture the spatial relationship of different regions on each scale of point clouds to extract discriminative local descriptors. An overview of our pyramid point transformer module is depicted in Fig. 1.

Like PointNet++ [38], our pyramid point transformer module has four transformer stages and thus generates the feature maps of the point clouds at four scales. Please note that the four transformer stages have the same structure, but the parameters are not shared. Given the point cloud with N points, we first construct the k-nearest neighbor (k-NN) graphs for the sampling points. We then feed each neighboring points to the graph embedding layer to extract local embedding of each sampling point. After that, the learned embeddings of the points are fed into the transformer E_1 to generate a new feature map $\boldsymbol{F}_1$ with the size of $\frac{N}{4^1} \times C_1$. Following this, using the feature map of the previous transformer stage E_{l-1} to the next transformer stage E_l, we can obtain a series of feature maps

$\{\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3, \boldsymbol{F}_4\}$ of the point clouds at four resolutions. Note that the number of the sampling points in the l-th stage is $\frac{N}{4^l}$. After performing the pyramid point transformer, the multi-scale spatial relationship between different regions of the point clouds can be captured so that the discrimination of the extracted local features can be enhanced.

Graph embedding. To characterize the local geometric structures of the point points, we construct the local k-NN graphs to extract the local embedding. Specifically, we first use the farthest point sampling (FPS) on the point cloud to sample the points as the center of each local neighborhood. Then, for each sampling point, we search for the k nearest points in the coordinate space to construct its local neighborhood. Compared with the query ball used in PointNet++ [38], the constructed local neighborhood can capture varying densities of different regions of the point clouds through the k-NN operation.

After that, we use graph convolution to extract local embedding of the point. in the point clouds. Assuming that we sample m points from the input points in the l-th stage as the seed points, we can obtain m neighborhoods, denoted by $\mathcal{N} = \{\mathcal{N}_i \in \mathbb{R}^{k \times (3+C)} \mid i = 1, \ldots, m\}$, where k is the number of neighbors and C indicates the C-dimensional feature of the point. Specifically, we use the EdgeConv [50] to characterize the local geometric structure of the local neighborhood of each point. Given the coordinate $\boldsymbol{p}_i \in \mathbb{R}^3$ and the feature $\boldsymbol{x}_i \in \mathbb{R}^C$ of the i-th seed point, the local feature is formulated as:

$$h_\Theta(\triangle \boldsymbol{p}_{ji}, \triangle \boldsymbol{x}_{ji}) = h_\Theta([\triangle \boldsymbol{p}_{ji}; \triangle \boldsymbol{x}_{ji}]) \tag{1}$$

where j indicates the index of the j-th neighboring points in $\mathcal{N}_i$, and $\triangle \boldsymbol{p}_{ji} = \boldsymbol{p}_j - \boldsymbol{p}_i$ and $\triangle \boldsymbol{x}_{ji} = \boldsymbol{x}_j - \boldsymbol{x}_i$ capture the difference of the local neighborhood in the coordinate and feature spaces, respectively. In Eq. (1), $[\cdot; \cdot]$ indicates the concatenation operation and $h_\Theta : \mathbb{R}^{3+C} \rightarrow \mathbb{R}^C$ is an embedding function. Finally, we use the max-pooling operation on the local k-NN graph to aggregate the local embedding of the point, which is defined as:

$$\boldsymbol{f}_i = \max_{j \in \mathcal{N}_i} h_\Theta(\triangle \boldsymbol{p}_{ji}, \triangle \boldsymbol{x}_{ji}) \tag{2}$$

Since the max-pooling operation is a symmetric function, the output $\boldsymbol{f}_i \in \mathbb{R}^C$ is invariant to the permutation of input point clouds. As a result, we can obtain a new feature map $\boldsymbol{F}_l \in \mathbb{R}^{m \times C}$ in the l-th stage after graph embedding.

Grouped self-attention. We develop a grouped self-attention to adaptively learn the spatial relationship between different regions of the point clouds to extract discriminative local features. Unlike the original self-attention used in [52], the proposed grouped self-attention is a lightweight but efficient version, which utilizes the grouping operation to enhance the discrimination of local features of the point clouds. Specifically, the architecture of the grouped self-attention is shown in Fig. 2. Given the feature map $\boldsymbol{F}_l \in \mathbb{R}^{m \times C}$ in the l-th stage after graph embedding, we first apply two group-wise 1×1 convolution on the feature map $\boldsymbol{F}_l$ to generate two feature maps, $i.e.$, the query map $\boldsymbol{Q}^{m \times C}$ and the key map $\boldsymbol{K}^{m \times C}$, respectively. At the same time, we apply another 1×1 convolution on the feature map $\boldsymbol{F}_l$ to generate the value map $\boldsymbol{V} \in \mathbb{R}^{m \times C}$. Assuming that the number of groups is G, we divide the query map $\boldsymbol{Q}$ along the channel direction into G groups, denoted by $\{\boldsymbol{Q}_g \in \mathbb{R}^{m \times C/G} \mid g = 1, \ldots, G\}$. Likewise, we divide the key map $\boldsymbol{K}$ into G groups, denoted by $\{\boldsymbol{K}_g \in \mathbb{R}^{m \times C/G} \mid g = 1, \ldots, G\}$. For the g-th group, the attention map $\boldsymbol{W}_g \in \mathbb{R}^{m \times m}$ is formulated by:

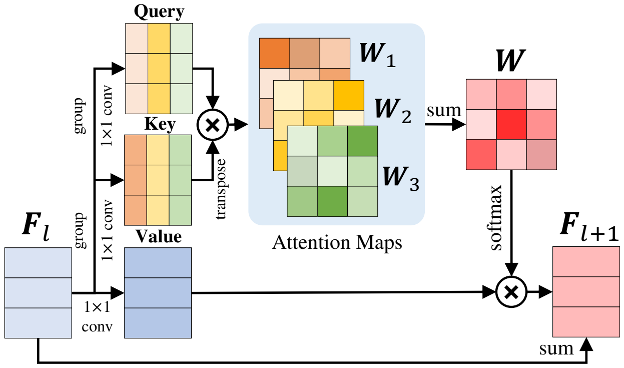


Figure 2: The overview of the grouped self-attention.

$$\boldsymbol{W}_g = \boldsymbol{Q}_g \cdot \boldsymbol{K}_g^\top \tag{3}$$

where $g \in [1, 2, \ldots, G]$ and $\boldsymbol{W}_g$ represents the similarity between the g-th query map $\boldsymbol{Q}_g$ and the g-th key map $\boldsymbol{K}_g$. Finally, we sum the G attention maps to obtain the final attention map $\boldsymbol{W}$, which is given below:

$$\boldsymbol{W} = \sum_{g=1}^{G} \boldsymbol{W}_g \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{m \times m}$ indicates the sum of the similarity between different regions in G groups. In this way, the points with similar neighboring structures can be assigned to large weights. Therefore, more discriminative local features of the points can be obtained. By multiplying the value map $\boldsymbol{V}$ with the attention map $\boldsymbol{W}$ followed by $softmax$ and adding the input embedding $\boldsymbol{F}_l$, the resulting feature map $\boldsymbol{F}_{l+1}$ is written as:

$$\boldsymbol{F}_{l+1} = softmax(\frac{\boldsymbol{W}}{\sqrt{C}})\boldsymbol{V} + \boldsymbol{F}_l \tag{5}$$

where C is the dimension of the query map and $\frac{1}{\sqrt{C}}$ is the scaling factor. The obtained new feature map $\boldsymbol{F}_{l+1} \in \mathbb{R}^{m \times C}$ is used as the input of the (l+1)-th stage. In addition, compared with the standard transformer encoder, we drop the position embedding because the coordinate of the point clouds already contains the position information. We also drop the feed-forward network for simplifying the network.

Compared with the original self-attention, the proposed grouped self-attention has the lower computational complexity. In Eqs. (3) and (4), the complexity of matrix multiplication and matrix addition is $O(m^2 \cdot C/G)$ and $O(m^2 \cdot G)$, respectively. Therefore, the total complexity of computing the attention map is $O(m^2 \cdot (C/G + G))$. However, the complexity of the original self-attention is $O(m^2 \cdot C)$. The ratio γ of the computational complexity of the grouped self-attention (GSA) and the original self-attention (SA) is defined as:

$$\gamma(GSA, SA) = \frac{m^2 \cdot (C/G + G)}{m^2 \cdot C} = 1/G + G/C \quad (6)$$

Generally, $C > G$, so the grouped self-attention has the lower computational complexity than the original self-attention. Note that if G=1, there is no need to perform the matrix sum, so $\gamma(GSA, SA) = (m^2 \cdot C/1)/(m^2 \cdot C) = 1$.

3.2. Pyramid VLAD

In order to obtain the discriminative global descriptor, we develop a pyramid VLAD module to aggregate multi-scale feature maps into the global descriptor for efficient retrieval. The architecture of our pyramid VLAD module is shown in Fig. 1.

Specifically, our pyramid VLAD module is built on multi-scale feature maps generated on point clouds with different spatial resolutions. In the pyramid point transformer, due to the different network depths, the generated feature maps of different resolutions have different representation capabilities. Therefore, we adopt a top-down structure to spread the high-level features into the low-level features to enhance the representation ability of the low-level features of the point clouds. Specifically, we define $\boldsymbol{F}_4^{'} = \boldsymbol{F}_4$ and the coordinate $\boldsymbol{F}_0 \in \mathbb{R}^{N \times 3}$ of the input point cloud in advance. Given the obtained four feature maps $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, $\boldsymbol{F}_3$, and $\boldsymbol{F}_4$, the top-down architecture can be formulated as:

$$\boldsymbol{F}_l^{'} = \mathrm{MLP}(\boldsymbol{F}_l \oplus \mathcal{I}(\boldsymbol{F}_{l+1}^{'})) \quad (7)$$

where $l \in \{0, 1, 2, 3\}$ and $\boldsymbol{F}_l{}' \in \mathbb{R}^{\frac{N}{4^l} \times C}$ is the generated new feature map. $\oplus$ represents the channel-wise concatenation, and $\mathcal{I}(\cdot)$ indicates the interpolation. Here, we adopt the distance based interpolation used in [38].

Based on the feature maps $\boldsymbol{F}_0^{'}$, $\boldsymbol{F}_1^{'}$, $\boldsymbol{F}_2^{'}$ and $\boldsymbol{F}_3^{'}$, we apply the VLAD pooling [1] to generate the global descriptors. For each feature map $\boldsymbol{F}_l^{'}$, the VLAD pooling learns K_l visual words, denoted as $\{\boldsymbol{c}_l^i \in \mathbb{R}^D \mid i = 1, \ldots, K_l\}$, and creates a $(D \times K_l)$-dimensional vector $\boldsymbol{V}_l = [\boldsymbol{V}_l^1, \ldots, \boldsymbol{V}_l^{K_l}]$. As a result, we can obtain four global descriptors of the point clouds at four resolutions.

After that, we use the context gating mechanism to aggregate the multi-scale descriptors into the discriminative global descriptor. Specifically, we first concatenate the multi-scale descriptors into a global descriptor. However, the obtained global descriptor is a high-dimensional vector, $i.e.$, a $(D \times \sum_{i=0}^{3} K_l)$-dimensional vector, which makes the query more time-consuming. To this end, we use a fully connected layer to compress the high-dimensional vector into a low-dimensional vector, denoted by $\boldsymbol{U}$. Then, we adopt the context gating mechanism on the low-dimensional vector to encode the global context information. The context gating mechanism converts the global descriptor $\boldsymbol{U}$ into a new global descriptor $\boldsymbol{U}^{'}$, which is formulated as:

$$\boldsymbol{U}^{'} = \sigma(\boldsymbol{W}\boldsymbol{U} + b) \circ \boldsymbol{U} \quad (8)$$

where $\boldsymbol{W} \in \mathbb{R}^{D \times D}$ and $b \in \mathbb{R}^D$ are trainable parameters, σ is the sigmoid activation and $\circ$ is the element-wise multiplication. Finally, we use the global descriptor $\boldsymbol{U}^{'}$ for efficient retrieval.

Dataset	Baseline		Refine	
	Training	Test	Training	Test
Oxford	21.7k	3.0k	21.7k	3.0k
In-house	-	4.5k	6.7k	1.7k

Table 1: Split of the baseline and refine datasets.

4. Experiments

4.1. Experimental Settings

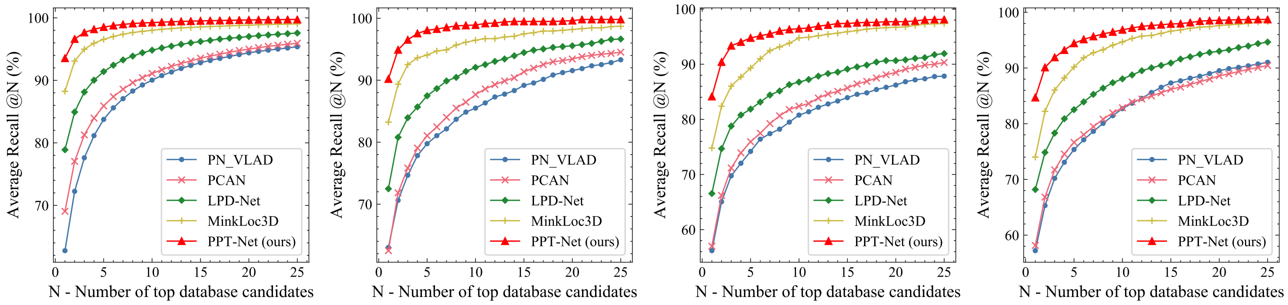
Benchmark datasets and evaluation. The benchmark datasets proposed in [47] are used for the evaluation of our method. It is built upon four open-source datasets, which include a partial set of Oxford RobotCar dataset [31] and three in-house datasets of the university sector (U.S.), residential area (R.A.), and business district (B.D.). All of them are collected by the LiDAR sensor mounted on a car that travels through these regions repeatedly at different times. The LiDAR scans are first organized into submaps that remove the non-informative ground planes and are downsampled to 4096 points. Moreover, all the submaps are tagged with a Universal Transverse Mercator (UTM) coordinate for ground-truth generating. During training, PointNetVLAD regards a point cloud pair with the distance less than 10m as a positive pair, and a point cloud pair with the distance greater than 50m as a negative pair. During testing, a point cloud retrieved from the database can be regarded a true match when the distance between the retrieved point cloud and the query point cloud is less than 25m. Following [47], we adopt two datasets: baseline and refine to evaluate our method. The detailed information of the baseline and refine datasets are shown in Tab. 1.

To evaluate the performance of place recognition, we use the evaluation metric *Recall@N* adopted in [47], which indicates the percentage of correctly matched queries. We report the average recall@1% (AR@1%) and average recall@1 (AR@1) metrics.

Implementation details. The architecture of our PPT-Net is shown in Fig. 1. We use the same number of LiDAR points as in PointNetVLAD [47]. In the experiment, we adopt a four-stage pyramid point transformer. The number of the constructed neighborhoods at each stage is 1024, 256, 64, and 16, respectively. When feeding the

Methods	Oxford		U.S.		R.A.		B.D.	
	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1
PN_VLAD [47]	80.9	62.6	72.7	63.2	60.8	56.1	65.3	57.2
PCAN [57]	83.9	69.4	79.1	62.4	71.1	56.9	66.9	58.1
LPD-Net [30]	91.0	80.9	85.7	72.6	78.9	66.7	74.9	68.3
LPD-Net [30] with h.f.	94.9	86.3	96.0	87.0	90.4	83.0	89.1	82.3
MinkLoc3D [23]	95.9	88.2	93.6	83.2	86.0	74.7	82.2	74.0
MinkLoc3D [23] with d.a.	97.9	**93.7**	95.0	86.0	91.1	81.1	88.4	82.6
PPT-Net (**ours**)	**98.1**	93.5	**97.5**	**90.1**	**93.3**	**84.1**	**90.0**	**84.6**

Table 2: Evaluation results of different place recognition methods trained on the **baseline dataset**. Note that "with h.f." means using the handcrafted feature, while "with d.a." means using data augmentation.



(a) Oxford

(b) U.S.

(c) R.A.

(d) B.D.

Figure 3: Average recall of different place recognition methods trained on the **baseline dataset**.

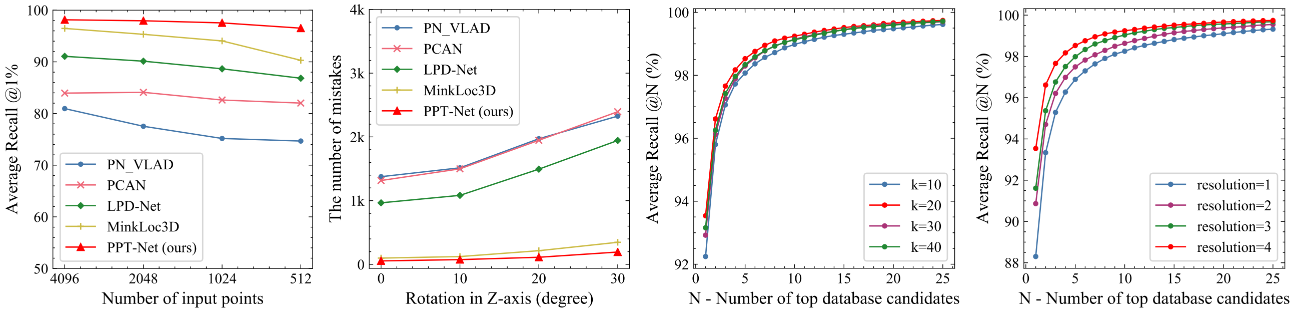
point cloud with the size of 4096×3 into the network, the neuron size of the obtained feature maps in four stages are 1024×64, 256×128, 64×256, and 16×512, respectively. In each transformer stage, we set the number of groups G to 8 in the grouped self-attention. After performing the pyramid point transformer on the point clouds, we use the pyramid VLAD module to aggregate local features into the discriminative global descriptors. We feed the obtained multi-scale point features into the pyramid VLAD module, and the neuron sizes are 64×256, 256×256, 1024×256, and 2048×256, respectively. In the pyramid VLAD module, the visual words of VLAD pooling from top to down are K_0=64, K_1=16, K_2=4, and K_3=1, respectively. The channel of these visual words are fixed to D=256. To train the proposed PPT-Net, we adopt the lazy quadruplet loss used in PointNetVLAD [47]. In addition, we use the PyTorch [35] to implement our method.

4.2. Place Recognition Results

Quantitative results. We compare the proposed PPT-Net with a series of advanced methods, including PointNetVLAD [47], PCAN [57], LPD-Net [30], DAGC [45], and MinkLoc3D [23]. Note that LPD-Net uses additional handcrafted features of the point clouds, while MinkLoc3D utilizes a data augmentation strategy during training. For a fair comparison, by running the official codes, we also report the results of their methods in the cases of not using handcrafted features or data augmentation. In addition, we denote PointNetVLAD as PN_VLAD for simplicity.

As shown in Tab. 2, we report the average recall@1% (AR@1%) and average recall@1 (AR@1) of different place recognition methods trained on the baseline dataset. From the table, it can be seen that our PPT-Net achieves the state-of-the-art on all datasets without using handcrafted features or data augmentation. Although some methods use handcrafted features ("LDP-Net with h.f.") or data augmentation ("MinkLoc3D with d.a.") to improve performance, our PPT-Net can still obtain good performance without using these tricks. For a fair comparison, our PPT-Net can further improve the performance of the Oxford dataset on the AR@1% metric from 95.9% to 98.1%. Compared with other methods, our method learns the spatial relationship of the point clouds by using the grouped self-attention on the local neighboring points to enhance the discrimination of the local features of the point clouds. In addition, we employ a pyramid VLAD to aggregate multi-scale feature maps into a discriminative global descriptor. What's more, the performance of three in-house datasets further demonstrate the generalization of our PPT-Net in the completely unseen environments. In Fig. 3, we also provide the recall curves of each method for the top 25 matches on four datasets. It can be clearly seen that our method is superior to the other methods.

In addition to the baseline dataset, we also evaluate our method on the refine dataset. As shown in Tab. 3, we report the AR@1% of different place recognition methods



(a) Different number of points (b) Different rotations (c) Different sizes (d) Different resolutions

Figure 4: (a) and (b): robustness analysis with different numbers of points and different angles of rotations. (c): Ablation study results of different k of the local k-NN graph in the pyramid point transformer. (d): Ablation study results of the point clouds with different resolutions in the pyramid VLAD.

Methods	Oxford	U.S.	R.A.	B.D.
PN_VLAD [47]	80.1	94.5	93.1	86.5
PCAN [57]	86.4	94.1	92.3	87.0
DAGC [45]	87.7	94.2	93.3	88.5
LPD-Net [30]	94.6	95.4	95.6	92.5
MinkLoc3D [23]	96.9	98.8	97.7	94.2
PPT-Net (**ours**)	**98.4**	**99.7**	**99.5**	**95.3**

Table 3: Evaluation results (AR@1%) of different place recognition methods trained on the **refine dataset**.

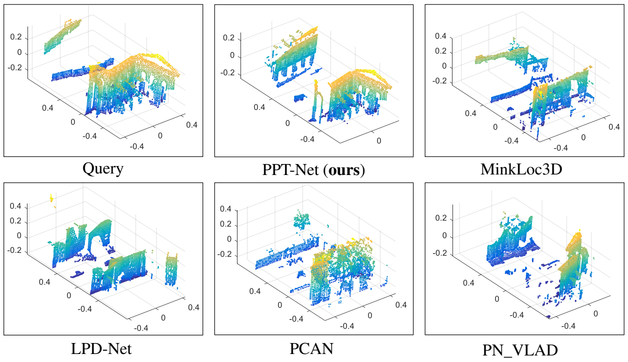


Figure 5: Example retrieval results of different methods.

on four datasets. It can be seen that the performance of our method outperforms other advanced methods by a large margin. The performance on both baseline and refine datasets further demonstrates the effectiveness of the proposed PPT-Net for point cloud based place recognition.

Visual results. We also visualize the query point clouds and the top 1 retrieved point clouds on the Oxford dataset with different place recognition methods, as shown in Fig. 5. It can be seen that compared with other advanced methods, our PPT-Net can successfully retrieve the correct match of the corresponding query point clouds.

Robustness analysis. For a more comprehensive evaluation, we conduct experiments on the baseline dataset to evaluate the robustness of the proposed PPT-Net. Specifically, we investigate the performance of different methods in terms of sparse scenes. In the experiment, we first randomly sample 4096 points into 2048, 1024, and 512 points, respectively. Then, we train the network on different number of points, respectively. As shown in Fig. 4(a), we report the test results of AR@1% and AR@1 on the Oxford dataset. It can be found that as the number of points decreases, the performance of our PPT-Net can still maintain the high performance. However, the performance of MinkLoc3D [23] decreases greatly. On the one hand, since MinkLoc3D is based on volumetric representation, compared with our point-based PPT-Net, it cannot capture the local fine geometric structures of the point clouds well. On the other hand, MinkLoc3D ignores the spatial relationship of the entire point clouds, so it cannot obtain discriminative local features. Thanks to the pyramid point transformer module, our PPT-Net can capture the multi-scale spatial relationship of the point clouds to generate discriminative local features of the point clouds. Therefore, our method can obtain better results in the cases of very sparse point clouds.

We also study the robustness of our PPT-Net to rotation. Specifically, we first train the network on the baseline dataset without rotation augmentation. Then, during the test, we rotate the input point cloud with 10°, 20°, and 30°, respectively. In Fig. 4(b), we show the total number of incorrectly retrieved scenes on the university dataset (U.S.) under different rotation degrees. If there is a wrong match at the top 1 retrieval, it is regarded as a wrong localization. It can be seen from the figure that when the degree pf rotation increases, the number of incorrectly retrieved scenes of our PPT-Net is less than that of other methods.

Computational cost. As shown in Tab. 4, we report the computation and memory requirements of different place recognition methods. Due to the pyramid structure and grouped operation, our PPT-Net has the lower trainable parameters compared with the point-based methods, including PointNetVLAD, PCAN, and LPD-Net. MinkLoc3D

Methods	Parameters	FLOPs	Running time per frame
PN_VLAD [47]	19.78M	4.2G	25ms
PCAN [57]	20.42M	7.7G	72ms
LPD-Net [30]	19.81M	7.8G	35ms
MinkLoc3D [30]	**1.10**M	**1.8**G	**21**ms
PPT-Net (**ours**)	13.12M	3.2G	22ms

Table 4: Evaluation of computation and memory requirements of different methods.

Models	AR@1%	AR@1
PointNet++ [38]	94.4	86.2
PPT-Net w/o Graph Embedding	96.0	90.5
PPT-Net w/o Transformer	97.3	92.1
PPT-Net	**98.1**	**93.5**

Table 5: Ablation studies of different components in the PPT-Net.

is a voxel-based method that adopts the MinkowskiEngine auto-differentiation library [5] for sparse tensors, so its trainable parameters are low. In addition, MinkLoc3D uses a simple generalized-mean pooling [32] to generate a global descriptor. Compared with VLAD pooling [1], it has few learnable parameters. For inference time, our PPT-Net is comparable with MinkLoc3D in terms of running time per frame (21 ms $vs.$ 22 ms). Since MinkLoc3D uses the sparse 3D CNNs in the network, the parameters and FLOPs are lower than others.

4.3. Ablation Study

Pyramid point transformer. We conduct the experiments on the baseline dataset to verify the effectiveness of our pyramid point transformer. As shown in Tab. 5, we report the AR@1% and AR@1 on the Oxford dataset. We compare the pyramid point transformer with the similar structure of PointNet++ [38]. It can be seen that our PPT-Net outperforms the PointNet++ with a large margin. In addition, we also study the impact of the local k-NN graph embedding and transformer on the performance. From the table, it can be found that simultaneously considering the local k-NN graph embedding and transformer can achieve the best performance.

Different k in local k-NN graph. We study the impact of different k in the local k-NN graph on the performance of place recognition. In Fig. 4(c), we show the recall curves of different k on the Oxford dataset. From the figure, it can be seen that our method achieves the best performance when k is set to 20. Since we use the pyramid point transformer module to enhance the discrimination of the local feature of the point clouds, the size k of the local k-NN graph has no distinct impact on retrieval performance.

Groups	AR@1%	AR@1	FLOPs
$G = 1$	97.5	92.6	3.8G
$G = 2$	97.9	92.9	3.6G
$G = 4$	98.0	93.1	3.4G
$G = 8$	**98.1**	**93.5**	3.2G
$G = 16$	97.4	92.0	**3.0**G

Table 6: Ablation study results of different number of groups G in the grouped self-attention.

Different G in grouped self-attention. Here we study the impact of the number of groups G in the grouped self-attention. As shown in Tab. 6, we report the AR@1%, AR@1 of our PPT-Net on the Oxford dataset. It can be found that G=8 achieves the best performance. Note that when G is set to 1, the grouped self-attention is the original self-attention. From the table, it can be found that the grouped self-attention can maintain the performance of self-attention while reducing the computational cost.

Different resolutions in pyramid VLAD. We conduct experiments to demonstrate the effectiveness of our pyramid VLAD module. Since our PPT-Net uses the point feature maps at four different resolutions, we study the impact of different resolutions on performance. In Fig. 4(d), we show the recall curves of the point clouds with different resolutions on the Oxford dataset. It can be found that our method can achieve the best performance when all four resolutions are used. Compared with the single-scale descriptor of the point clouds, the multi-scale descriptors can capture different levels of varying density of the point clouds. Therefore, we can obtain more discriminative global descriptors for efficient retrieval.

5. Conclusion

In this paper, we proposed a novel pyramid point cloud transformer network (PPT-Net) for point cloud based place recognition. In order to extract the discriminative local features of point clouds, we proposed the pyramid point transformer module that uses the grouped self-attention on multi-scale regions of point clouds to characterize the spatial relationship. In order to aggregate the local features of the point clouds into a discriminative global descriptor, we developed the pyramid VLAD module that uses the context gating mechanism to aggregate the multi-scale descriptors generated by the VLAD pooling into a discriminative global descriptor. Extensive experiments on the Oxford dataset and three in-house datasets can demonstrate the effectiveness of the proposed method for the point cloud based place recognition task.

Acknowledgments

This work was supported by the National Science Fund of China (Grant Nos. U1713208, 61876084).

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016.

[2] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

[3] Fengkui Cao, Yan Zhuang, Hong Zhang, and Wei Wang. Robust place recognition and loop closing in laser-based SLAM for ugvs in urban environments. *IEEE Sensors Journal*, 18(10):4242–4252, 2018.

[4] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. SSPC-Net: Semi-supervised semantic 3D point cloud segmentation network. In *AAAI*, 2021.

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019.

[6] Konrad P Cop, Paulo VK Borges, and Renaud Dubé. DELIGHT: An efficient descriptor for global localisation using LiDAR intensities. In *ICRA*, 2018.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[8] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *ECCV*, 2018.

[9] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *CVPR*, 2018.

[10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3D local features for direct pairwise registration. In *CVPR*, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Juan Du, Rui Wang, and Daniel Cremers. DH3D: Deep hierarchical 3D descriptors for robust large-scale 6DoF relocalization. In *ECCV*, 2020.

[14] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[15] Andrea Frome, Daniel Huber, Ravi Krishna Kolluri, Thomas Bulow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[16] Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. Texture mapping for 3D reconstruction with rgb-d sensor. In *CVPR*, 2018.

[17] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Wieser Andreas. The perfect match: 3D point cloud matching with smoothed densities. In *CVPR*, 2019.

[18] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. PCT: Point cloud transformer. *arXiv preprint arXiv:2012.09688*, 2020.

[19] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, and Marc Pollefeys. 3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. *Image and Vision Computing*, 68:14–27, 2017.

[20] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *CVPR*, 2013.

[21] Le Hui, Mingmei Cheng, Jin Xie, and Jian Yang. Efficient 3D point cloud feature learning for large-scale place recognition. *arXiv preprint arXiv:2101.02374*, 2021.

[22] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[23] Jacek Komorowski. MinkLoc3D: Point cloud based large-scale place recognition. In *WACV*, 2021.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[25] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[26] Liu Liu, Hongdong Li, Yuchao Dai, and Quan Pan. Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2432–2444, 2017.

[27] Zhe Liu, Weidong Chen, Junguo Lu, Hesheng Wang, and Jingchuan Wang. Formation control of mobile robots using distributed controller with sampled-data and communication delays. *IEEE Transactions on Control Systems Technology*, 24(6):2125–2132, 2016.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[29] Zhe Liu, Chuanzhe Suo, Yingtian Liu, Yueling Shen, Zhijian Qiao, Huanshu Wei, Shunbo Zhou, Haoang Li, Xinwu Liang, Hesheng Wang, et al. Deep learning-based localization and perception systems: approaches for autonomous cargo transportation vehicles in large-scale, semiclosed environments. *IEEE Robotics and Automation Magazine*, 27(2):139–150, 2020.

[30] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yunhui Liu. LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, 2019.

[31] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[32] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.

[33] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[34] Liyuan Pan, Yuchao Dai, Miaomiao Liu, and Fatih Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *CVPR*, 2017.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017.

[37] Charles R Qi, Hao Su, Matthias Niebner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *CVPR*, 2016.

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[39] Gernot Riegler, Ali Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, 2017.

[40] Timo Röhling, Jennifer Mack, and Dirk Schulz. A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In *IROS*, 2015.

[41] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, 2009.

[42] Radu Bogdan Rusu, Nico Blodow, Zoltancsaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *IROS*, 2008.

[43] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[44] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, 2015.

[45] Qi Sun, Hongyan Liu, Jun He, Zhaoxin Fan, and Xiaoyong Du. DAGC: Employing dual attention and graph convolution for point cloud based place recognition. In *ICMR*, 2020.

[46] Hugues Thomas, Charles R Qi, Jeanemmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019.

[47] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, 2018.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[49] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[50] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015.

[52] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *CVPR*, 2021.

[53] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

[54] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[55] Zi Jian Yew and Gim Hee Lee. 3DFeat-Net: Weakly supervised local 3D features for point cloud registration. In *ECCV*, 2018.

[56] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3DMatch: Learning the matching of local 3D geometry in range scans. In *CVPR*, 2017.

[57] Wenxiao Zhang and Chunxia Xiao. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. In *CVPR*, 2019.

[58] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020.