SOE-Net: A Self-Attention and Orientation Encoding Network for Point Cloud based Place Recognition

Yan Xia[1] Yusheng Xu[1†] Shuang Li[2] Rui Wang[1†] Juan Du[1] Daniel Cremers[1] Uwe Stilla[1]
[1]Technical University of Munich [2]Beijing Institute of Technology
{yan.xia, yusheng.xu, stilla}@tum.de, {wangr, duj, cremers}@in.tum.de, shuangli@bit.edu.cn

Abstract

We tackle the problem of place recognition from point cloud data and introduce a self-attention and orientation encoding network (SOE-Net) that fully explores the relationship between points and incorporates long-range context into point-wise local descriptors. Local information of each point from eight orientations is captured in a PointOE module, whereas long-range feature dependencies among local descriptors are captured with a self-attention unit. Moreover, we propose a novel loss function called Hard Positive Hard Negative quadruplet loss (HPHN quadruplet), that achieves better performance than the commonly used metric learning loss. Experiments on various benchmark datasets demonstrate superior performance of the proposed network over the current state-of-the-art approaches. Our code is released publicly at https://github.com/Yan-Xia/SOE-Net.

1. Introduction

Place recognition and scene localization in large-scale and complex environments is a fundamental challenge with applications ranging from autonomous driving [13, 14, 22] and robot navigation [11, 29] to augmented reality [19]. Given a query image or a LiDAR scan, the aim is to recover the closest match and its location by traversing a pre-built database. In the past decade, a variety of image-retrieval based solutions have shown promising performance [18, 20, 21]. However, the performance of image-based methods often degrades when facing drastic variations in illumination and appearance caused by weather and seasonal changes [1]. As a possible remedy, 3D point clouds acquired from LiDAR offer accurate and detailed 3D information that is inherently invariant to illumination changes. As a consequence, place recognition from point cloud data is becoming an increasingly attractive research topic. Fig. 1 (Top) shows a typical pipeline for point cloud based place recognition: One first constructs a database with LiDAR scans tagged with UTM coordinates acquired from GPS/INS readings. Given a query LiDAR scan, we then retrieve the closest match and its corresponding location from the database.

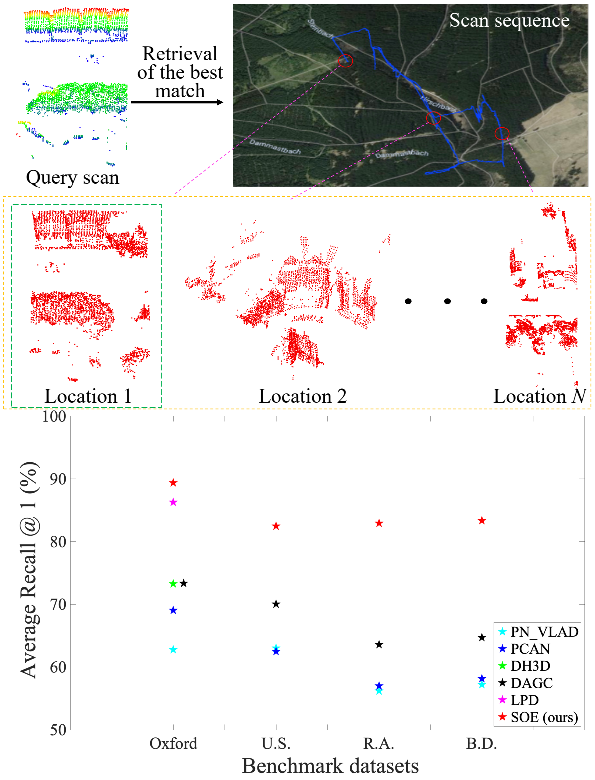


Figure 1. (Top) Place recognition from 3D point clouds: the street scene of a route map (shown in blue line) is denoted by a set of real-scan point clouds tagged with UTM coordinates. Given a query scan, we retrieve the closest match in this map (shown in the green box) to get the location of the query scan. (Bottom) Comparing the average recall at top 1 retrieval, SOE-Net significantly outperforms all published methods on various datasets.

The main challenge of point cloud based place recognition lies in how to find a robust and discriminative global

[†]Corresponding author.

descriptor for a local scene point cloud. While there exist abundant works on learning image descriptors, learning from point cloud data is far less developed. To date, only a few networks have been proposed for point cloud based place recognition in large-scale scenarios. PointNetVLAD [1] is a pioneering work, which first extracts the local features from 3D point clouds using PointNet [31] and then fuses them into global descriptors using the NetVLAD [2] layer. PCAN [40] adapts the PointNet++ [32] architecture to generate an attention map which re-weights each point during the local descriptors aggregation stage. Both methods use PointNet [31] to extract local descriptors, which ignores the geometric relationship among points. As of late, the authors of DH3D [8] and DAGC [36] noticed this shortcoming and designed advanced local feature description networks. While this results in better local descriptors, both approaches simply aggregate these local descriptors to a global descriptor using the PCAN or PointNetVLAD fusion architecture, without considering the long-range dependencies of different features.

Similar to these previous studies [8, 36], we also notice the importance of better utilizing the neighborhood context of each point when extracting its geometric representation. To tackle this problem, we adopt a point orientation encoding (PointOE) module to encode the neighborhood information of various orientations. The orientation-encoding unit is integrated with PointNet [31], taking its advantages in feature representation learning provided by the multi-layer perceptrons. Another observation is, when aggregated into a global descriptor, different local descriptors should tactically contribute unevenly. To achieve this, we develop a self-attention unit to introduce long-range contextual dependencies, encoding the spatial relationships of the local descriptors for weighting. Combining the principals above, we propose a novel network named SOE-Net (Self-attention and Orientation Encoding Network). It is an end-to-end architecture that explores the relationship among the raw 3D points and the different importance of local descriptors for large-scale point cloud based retrieval. Specifically, SOE-Net combines local descriptor extraction and aggregation, which enables one-stage training to generate a discriminative and compact global descriptor from a given 3D point cloud. Additionally, we propose a novel "Hard Positive Hard Negative quadruplet" (HPHN quadruplet) loss, which addresses some of the limitations of the widely used lasy quadruplet loss. To summarize, main contributions of this work include:

- We propose a novel point orientation encoding (PointOE) module to effectively extract local descriptors from a given point cloud, considering the relationship between each point and its neighboring points. We further design a self-attention unit to differentiate the importance of different local descriptors to a global descriptor.
- We present a new loss function termed HPHN quadruplet loss that is more effective for large-scale point cloud based retrieval. Comparing with previous loss functions, it can achieve more versatile global descriptors by relying on the maximum distance of positive pairs and the minimum distance of negative pairs.
- We conduct experiments on four benchmark datasets, including Oxford RobotCar [28] and three in-house datasets to demonstrate the superiority of SOE-Net over other state-of-the-art methods. Notably, the performance on Oxford RobotCar reaches a recall of 89.37% at top 1 retrieval.

2. Related work

Usually, the implementation of place recognition based on 3D point cloud retrieval is converted to a 3D feature matching problem, in which the 3D descriptor has substantial impact on the matching performance. Numerous methods for extracting 3D descriptors from point clouds have been developed, which can be roughly grouped into two categories: local descriptors and global descriptors.

3D local descriptors. Encoding robust local geometric information has long been a core challenge in robotics and 3D vision, with various attempts made. For example, extracting local structural information as histograms is representative. Spin Image (SI) [16] deploys the spin image representation to match 3D points. Geometry Histogram [10] proposes a novel regional shape context descriptor to improve the 3D object recognition rate on noisy data. Point Feature Histograms (PFH) [35] and Fast Point Feature Histograms (FPFH) [34] seek to calculate the angular features and surface normals to represent the relationship between a 3D point and its neighbors. However, these handcrafted descriptors are unsuitable for large-scale scenarios due to the computational cost, at the mean time which are also sensitive to noisy and incomplete data acquired by sensors. Recently, learning-based methods for 3D local descriptor extraction have gained significant developments boosted by large-scale 3D datasets. 3DMatch [38] converts 3D points to voxels and then uses a 3D convolution network for segment matching. PPFNet [7] and PPF-FoldNet [6] directly use the raw 3D points as input and learn point pair features from points and normals of local patches. Fully Convolutional Geometric Features (FCGF) [5] proposes a compact geometric feature computed by a 3D fully-convolutional network. 3DFeatNet [37] designs a weakly supervised network to learn both the 3D feature detector and descriptor. Furthermore, ASLFeat [27] focuses on mitigating limitations in the joint learning of 3D feature detectors and descriptors. 3DSmoothNet [12] and DeepVCP [26] learn compact and rotation invariant 3D descriptors relying on 3D CNNs. RSKDD-Net [25] introduces random sampling concept to efficiently learn keypoint detector and descriptor. Some methods explore to compress the dimensions of handcrafted 3D local descriptors utilizing deep learning, such as Compact Geometric Features (CGF) [17] and LORAX [9].

3D global descriptors. Different from 3D local descriptors, 3D global descriptors encapsulate comprehensive and global information of the entire scene. Most handcrafted global descriptors describe places with global statistics of LIDAR scans. [33] proposes a fast method of describing places through histograms of point elevation. DELIGHT [30] designs a novel global descriptor by leveraging intensity information of LiDAR data. [3] converts 3D points to 2D images and then extracts ORB features from these images for scene correspondence. With breakthroughs of learning based image retrieval methods, deep learning on 3D global descriptors for retrieval tasks has drawn growing attention. PointNetVlad [1] first tackles 3D place recognition in an end-to-end way, which combines PointNet [31] and NetVlad [2] to extract global descriptors from 3D points. Following this, PCAN [40] proposes an attention mechanism for local features aggregation, discriminating local features that contribute positively. However, these two methods employ PointNet architecture for extracting local features, which does not particularly concern the local geometry. LPD-Net [23] extracts the local contextual relationships but relies on handcrafted features . DH3D [8] designs a deep hierarchical network to produce more discriminative descriptors. DAGC [36] introduces a graph convolution module to encode local neighborhood information. However, it does not count the spatial relationship between local descriptors. Compared with previous studies, our network combines the strengthens of their designs, facilitating discriminative and versatile global descriptors.

3. Problem Statement

Let M_{ref} be a pre-built reference map of 3D point clouds defined with respect to a fixed reference frame, which is divided into a set of submaps such that $M_{ref} = \{m_i : i = 1, ..., M\}$. The submap coverages are kept approximately the same. Each submap is tagged with a UTM coordinate at its centroid using GPS/INS. Let Q be a query point cloud with the same coverage with respect to a submap in M_{ref}. We define the place recognition problem as retrieving a submap m^* from M_{ref} that is structurally closest to Q. Note that under this formulation, Q is not a subset of M_{ref}, since they are independently scanned at different times.

To tackle this problem, we design a neural network to learn a function $f(\cdot)$ that embeds a local point cloud to a 3D global descriptor of pre-defined size. The goal is to find a submap $m^* \in M_{ref}$ such that the distance between global descriptors $f(m^*)$ and $f(Q)$ is minimized:

$$m^* = \underset{m_i \in M_{ref}}{\operatorname{argmin}}\ d(f(Q), f(m_i)), \qquad (1)$$

where $d(\cdot)$ is a distance metric (e.g., Euclidean distance). In practical implementation, a global descriptor dictionary is built offline for all 3D submaps. When a query scan appears, the nearest submap is obtained efficiently by comparing the global descriptor extracted online from the query scan with stored global descriptors.

4. SOE-Net

Fig. 2 shows the overall network architecture of our SOE-Net, where the local descriptor extraction part produces local descriptors from the 3D query scan, and the descriptor aggregation part aims to generate a distinct global descriptor.

Given the input as a query point cloud with coordinates denoted as $Q = \{p_1, \cdots, p_N\} \in \mathbb{R}^{N\times3}$, we first use the designed PointOE module to extract point-wise local descriptors. Unlike previous studies, it extracts relevant local information from eight directions to enhance point-wise feature representation, with details described in Section 4.1.1. Then we propose a self-attention unit in the descriptor aggregation part to encode the spatial relationship among point-wise local descriptors, which is explained in Section 4.2.1. Afterwards, the NetVLAD layer is adopted to fuse enhanced local descriptors in Section 4.2.2. The training strategy with the proposed HPHN quadruplet loss is presented in Section 4.3.

4.1. Local descriptor extraction

4.1.1 PointOE Module

The successes of many non-learning based image retrieval methods are owing to the design of great local image descriptors (e.g., SIFT [24]). Orientation-encoding (OE) is one of SIFT's most shining highlights, which is also considered to benefit 3D feature description. Inspired by PointSIFT [15], we introduce the OE unit to the proposed SOE-Net. Specifically, we integrate it into PointNet to improve the point-wise feature representation ability. Fig. 3 shows the detailed architecture of the PointOE module. To the best of our knowledge, no prior work has explored it for large-scale place recognition and its effectiveness for retrieval has not been verified.

The inputs to our PointOE module are the 3D coordinates of N points. Following [31], multi-layer perceptrons (MLP) are adapted to encode the input 3D coordinates into features of $[64, 128, 256, 1024]$ dimensions. We insert the OE unit in front of each MLP to improve the representation ability. Local descriptors F_L are generated from this module.

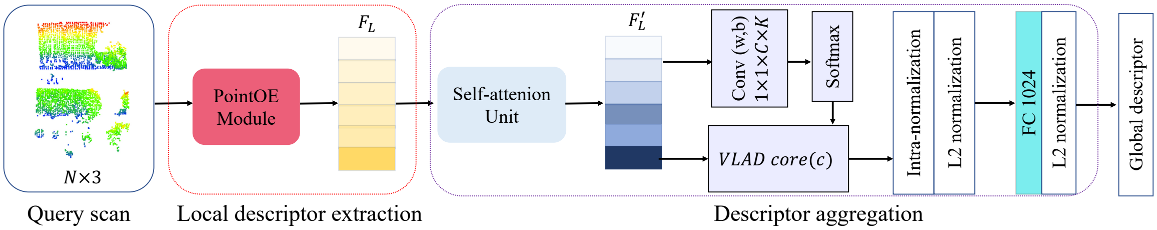


Figure 2. Overview of the SOE-Net architecture. The network takes a query scan with N points as input and employs the PointOE module to extract point-wise local descriptors F_L. During descriptor aggregation, a self-attention unit is applied on the local descriptors and followed by the NetVLAD layer. Finally, a fully connected (FC) layer is adopted to compress the output descriptor vector, follow by the L2 normalization to produce a global descriptor.

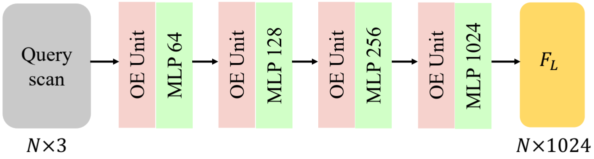


Figure 3. PointOE module. The input point cloud passes through a series of OE units and MLPs, local descriptors F_L are generated as output.

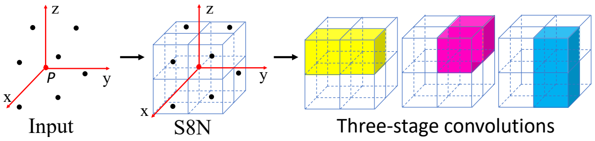


Figure 4. Illustration of OE unit.

Orientation-encoding Unit. Consider a $N \times C$ matrix as an input which describes a point cloud of size N with a C-dimensional feature for each point, OE unit will output a feature map with the same dimension $N \times C$. Every point is assigned to a new C-dimensional feature, which integrates the local information from eight orientations. As shown in Fig. 4, the OE unit first adopts the Stacked 8-neighborhood Search (S8N) to find the nearest neighbors for point P in each of the eight octants [15]. Furthermore, we extract features using three-stage convolutions from those neighbors, which lie in a $2 \times 2 \times 2$ cube along the $x-, y-, z-$ axis. Formally, these three-stage convolutions are defined as:

$$\begin{aligned} OE_x &= \mathrm{ReLU}(\mathrm{Conv}(w_x, V, b_x)), \\ OE_{xy} &= \mathrm{ReLU}(\mathrm{Conv}(w_y, OE_x, b_y)), \\ OE_{xyz} &= \mathrm{ReLU}(\mathrm{Conv}(w_z, OE_{xy}, b_z)), \end{aligned} \tag{2}$$

where $V \in \mathbb{R}^{2\times2\times2\times C}$ are the feature vectors of neighboring points. $w_x \in \mathbb{R}^{2\times1\times1\times C}, w_y \in \mathbb{R}^{1\times2\times1\times C}$ and $w_z \in \mathbb{R}^{1\times1\times2\times C}$ are weights of the three-stage convolutions, b_x, b_y, b_z are the biases of convolution operators. By this way, the OE unit captures the local geometric structure from eight spatial orientations.

4.2. Feature Aggregation

4.2.1 Self-attention Unit

To introduce long-range context dependencies after extracting local descriptors, we design a self-attention unit [39] before fusing them into the NetVLAD layer. The self-attention unit can encode meaningful spatial relationships between local descriptors. Fig. 5 presents its architecture. Given local descriptors $F_L \in \mathbb{R}^{N\times C}$, where N is the number of points and C is the number of channels, we feed F_L into two MLPs respectively and generate the new feature maps $X \in \mathbb{R}^{N\times C}$, $Y \in \mathbb{R}^{N\times C}$. Then the attention map W is calculated, defined as follows:

$$W_{j,i} = \frac{\exp(Y_j \cdot X_i^T)}{\sum_{i,j=1}^{N} \exp(Y_j \cdot X_i^T)}, \tag{3}$$

where $W_{j,i}$ indicates that the i^{th} local descriptor impacts on j^{th} local descriptor, with the shape of $N \times N$. Here, it deems as the component that learns the long-range dependency relationship among point-wise local descriptors. More important local descriptors will contribute more to the representation of the target global descriptor. On the other hand, F_L is fed into another MLP to output a new feature map $Z \in \mathbb{R}^{N\times C}$. Afterwards, we multiply it with the transpose of W to generate the result $A^P \in \mathbb{R}^{N\times C}$. Finally, we add a scale parameter α on it and add back F_L, which can be defined as follows:

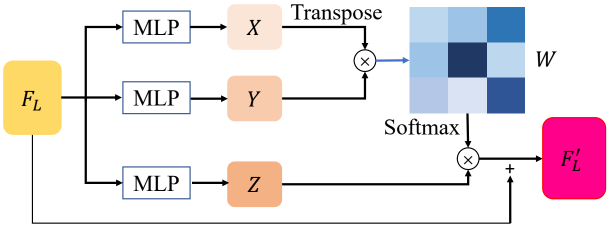


Figure 5. Illustration of the self-attention unit.

$$F_L^{'} = \mu A^p + F_L = \mu W^T Z + F_L, \tag{4}$$

where μ is initialized as zero and gradually assigned more weights with the progress of learning. The final output has a global context view compared with the original local descriptors. This enhances feature integration by combining geometrical and contextual information.

4.2.2 NetVLAD Layer

In this module, we aim to aggregate the local descriptors to a discriminative and compact global one. Following the configuration in [1], we adopt a NetVLAD layer to fuse features. The NetVLAD layer learns K visual words, denoted as $\{v_1, \cdots, v_K | v_k \in \mathbb{R}^C\}$, and generates a $(C \times K)$-dimensional VLAD descriptor $F_{VLAD} = \{F_{vlad}^1, \cdots, F_{vlad}^K\}$. However, the VLAD descriptor is time-consuming for nearest neighbor search, thus we apply a fully connected layer to generate a more compact global descriptor with an L_2 normalization.

4.3. Loss function

Before going to the details of our proposed HPHN-quadruplet loss, we give a short review on the quadruplet loss [4] and its improvement. To compute the quadruplet loss, each batch of the training data includes T quadruplets. Each quadruplet is denoted as $\Gamma_q = (\delta_a, \delta_p, \delta_n, \delta_n^*)$, where δ_a is a anchor point cloud, δ_p a positive point cloud (structurally similar to the query), δ_n a negative point cloud (structurally dissimilar to the query), δ_n^* a randomly sampled point cloud that is different with $\delta_a, \delta_p, \delta_n$. The quadruplet loss is formulated as:

$$L_q = \frac{1}{T}\sum^{T}\left[||f(\delta_a)-f(\delta_p)||_2^2-||f(\delta_a)-f(\delta_n)||_2^2+\alpha\right]_+ \\ +\frac{1}{T}\sum^{T}\left[||f(\delta_a)-f(\delta_p)||_2^2-||f(\delta_n^*)-f(\delta_n)||_2^2+\beta\right]_+, \tag{5}$$

where $[\cdots]_+$ denotes the hinge loss, α and β are the constant margins. The first term is a triplet loss which focuses on maximizing the feature distance between the anchor point cloud and the negative point cloud. The second term focuses on maximizing the feature distance between the negative point cloud and the additional point cloud δ_n^*.

To make the positive and negative samples in the quadruplet more effective, the quadruplet loss is extended to the lazy quadruplet loss [1] by introducing hard sample mining. The quadruplets now become $\Gamma_{lq} = (\delta_a, \{\delta_p\}, \{\delta_n\}, \delta_n^*)$, where $\{\delta_p\}$ is a collection of ϕ positive point clouds and $\{\delta_n\}$ is a collection of φ negative point clouds. The loss is modified accordingly to

$$L_{lq} = \max_{\substack{i\,=\,1\ldots\phi \\ j\,=\,1\ldots\varphi}}\left(\left[||f(\delta_a)-f(\delta_p^i)||_2^2-||f(\delta_a)-f(\delta_n^j)||_2^2+\alpha\right]_+\right) \\ +\max_{\substack{i\,=\,1\ldots\phi \\ j\,=\,1\ldots\varphi}}\left(\left[||f(\delta_a)-f(\delta_p^i)||_2^2-||f(\delta_n^*)-f(\delta_n^j)||_2^2+\beta\right]_+\right). \tag{6}$$

In practice, a common strategy is to set β to be smaller than α (e.g., $\alpha = 0.5, \beta = 0.2$) to make the second term in Eq. 6 a relatively weaker constraint. However, we find this practice is less justified, especially in the scenario of metric learning for large-scale place recognition. In this work, we propose the Hard Positive Hard Negative quadruplet loss (HPHN quadruplet), which unifies the margin selection for δ_a and δ_n^*, and meanwhile rely on the hardest positive and the hardest negative samples in the batch to compute the learning signal. In our case, the hardest positive point cloud δ_{hp} is the least structurally similar to the anchor point cloud, which is defined as:

$$\delta_{hp} = \underset{\delta_p^i \in \{\delta_p\}}{\operatorname{argmax}}\ ||f(\delta_a) - f(\delta_p^i)||_2^2, \tag{7}$$

The hardest negative point cloud is the most structurally dissimilar to the anchor point cloud. Here, we first find the hard negative point cloud δ_{hn} in $\{\delta_n\}$, which is defined as:

$$\delta_{hn} = \underset{\delta_n^j \in \{\delta_n\}}{\operatorname{argmin}}\ ||f(\delta_a) - f(\delta_n^j)||_2^2. \tag{8}$$

Additionaly, we consider the feature distance from δ_n^* to δ_n:

$$\delta_{hn}^{'} = \underset{\delta_n^j \in \{\delta_n\}}{\operatorname{argmin}}\ ||f(\delta_n^*) - f(\delta_n^j)||_2^2. \tag{9}$$

Finally, we select one of them as the hardest negative training data, which has the minimum distance d_{hn}:

$$d_{hn} = \min(||\mathrm{f}(\delta_\mathrm{a}) - \mathrm{f}(\delta_\mathrm{hn})||_2^2, ||\mathrm{f}(\delta_\mathrm{n}^*) - \mathrm{f}(\delta_\mathrm{hn}^{'})||_2^2). \tag{10}$$

In conclusion, the HPHN quadruplet loss can be formulated as:

$$L_{HPHN} = \left[||f(\delta_a) - f(\delta_{hp})||_2^2 - d_{hn} + \gamma\right]_+, \tag{11}$$

where γ is the unified margin. The first term in Eq. 11 is the upper bound of the feature distance of all the positive point cloud pairs, and the second term is the lower bound of the feature distance of all the negative point cloud pairs in a batch.

Although having a form similar to the triplet loss, our loss is still a quadruplet loss that is computed from the sampled quadruplet. Compared with the lazy quadruplet loss, the proposed HPHN quadruplet loss picks the harder term between Eq. 8 and Eq. 9, instead of using both in the loss computation. Moreover, the same margin is used when either of the both is selected. Despite this simple modification, our experimental results in Section 6.1 demonstrate that our HPHN quadruplet loss significantly outperforms the lazy quadruplet loss.

4.4. Implements Details

The proposed network is implemented in the Tensorflow framework and trained on a single Nvidia Titan Xp GPU with 12G memory. The size of the input points is 4096. The margins γ for the HPHN quadruplet loss are set to 0.5. Similar to all previous methods, we set the number of clusters K in the NetVLAD layer to 64. In the training stage, we set the batch size to 1 in each training iteration. Adam optimizer is used in the models for epoch 20. Same as PCAN, we choose 2 positive point clouds and 9 negative point clouds (including 1 other negative point cloud) in caculating loss functions. The initial learning rate is set to 0.0005. It is decayed by 0.7 after every 200K steps.

5. Experiments

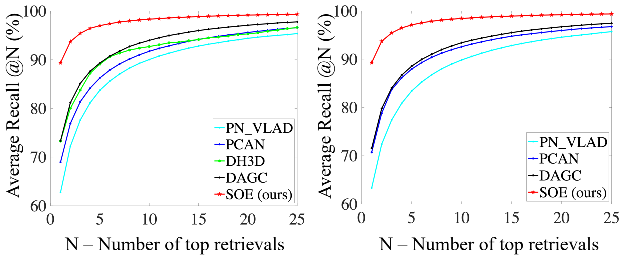
5.1. Benchmark Datasets

We evaluate the proposed method on benchmark datasets proposed in [1]. It includes four open-source datasets for different scenes: Oxford RobotCar [28] dataset and three in-house datasets of a university sector (U.S.), a residential area (R.A.), and a business district (B.D.). These datasets are all collected by a LiDAR sensor mounted on a car that travels around the regions repeatedly at different times. Based on the LiDAR scans, a database of submaps is built and each submap is tagged with a UTM coordinate. To better learn geometric features, the non-informative ground planes of all reference submaps are removed. The size of each submap is downsampled to 4096 points. In training, point clouds are regarded as correct matches if they are at maximum 10 m apart and wrong matches if they are at least 50 m apart. In testing, we regard the retrieved point cloud as a correct match if the distance is within 25 m between the retrieved point cloud and the query scan. Following the experimental settings in [1, 40, 36], we first train our method using only the Oxford RobotCar training dataset. We will henceforth refer to this as our *Baseline Network*. To improve the generalizability of the network, we further add the U.S. and R.A. training datasets into training data as our *Refinement Network*.

5.2. Results

5.2.1 Baseline Network

We compare our baseline network with PointNetVLAD (PN_VLAD) [1] as a baseline and the state-of-the-art methods PCAN [40], LPD-Net [23], DH3D [8], and DAGC [36]. For a fair comparison, we use the same evaluation metrics, including the Average Recall at Top N and Average Recall at Top 1%. The final global descriptors of all networks are 256-dim. Table 1 shows the top 1% recall of each network on the four datasets. We refer to the recall values reported in [8, 36, 23, 40, 1]. The recall values of DH3D for U.S., R.A. and B.D. are not reported in [8].



(a) Baseline (b) Refinement

Figure 6. Average recall of SOE-Net tested on Oxford RobotCar. (a) shows the average recall when SOE-Net is only trained on Oxford RobotCar. (b) shows the average recall when SOE-Net is trained on Oxford RobotCar, U.S. and R.A. DH3D is not trained on this dataset in [8].

	SOE	DAGC	DH3D	LPD	PCAN	PN_VLAD
Oxford	**96.40**	87.49	84.26	94.92	83.81	80.31
U.S.	93.17	83.49	-	**96.00**	79.05	72.63
R.A.	**91.47**	75.68	-	90.46	71.18	69.75
B.D.	88.45	71.21	-	**89.14**	66.82	65.30

Table 1. The average recall (%) at top 1% for each network.

The results show that the proposed baseline network outperforms others significantly on Oxford RobotCar dataset. The best performance on Oxford RobotCar reaches the recall of 96.40 % at top 1%, exceeding the recall of current state-of-the-art method LPD-Net by 1.52 %. Furthermore, SOE-Net achieves the recall of 93.17%, 91.47%, 88.45% on the unseen datasets respectively, which is similar or slightly weaker than LPD-Net. However, both of them improve the performance by a large margin compared with other methods. Notably, LPD-Net relies on ten handcrafted features, which has complex network architecture and high computational cost. Fig. 6 (a) shows the recall curves of PointNetVLAD, PCAN, DAGC, and SOE-Net for the top 25 retrieval results. Notably, the recall at top 1 of SOE-Net reaches a recall of 89.37%, indicating the proposed network effectively captures the task-relevant local information and generate more discriminative global descriptors. More qualitative results are given in Section 5.4.

5.3. Refinement Network

To improve the generalizability of the network on the unseen scenarios, [40, 1, 36] further add U.S. and R.A. to the training data. We follow the same training sets to train our refinement network. As illustrated in Table 2, SOE-Net still significantly outperforms the state-of-the-art method DAGC on all datasets. By comparing Table 1 and Table 2, it becomes clear that adding more data from different scenarios improves the performance of SOE-Net on the unseen dataset B.D.. In other words, given more publicly accessible datasets of real scans, SOE-Net has huge potential for

LiDAR based localization. In Fig. 6 (b) we plot the recall curves of the refinement network of PointNetVLAD, PCAN, DAGC, and SOE-Net for the top 25 retrieval results. It demonstrates that the global descriptors generated by SOE-Net are more discriminative and generalizable than all previously tested state-of-the-art methods.

	Ave recall @1%			Ave recall @1		
	SOE	DAGC	PCAN	SOE	DAGC	PCAN
Oxford	**96.43**	87.78	86.40	**89.28**	71.39	70.72
U.S.	**97.67**	94.29	94.07	**91.75**	86.34	83.69
R.A.	**95.90**	93.36	92.27	**90.19**	82.78	82.26
B.D.	**92.59**	88.51	87.00	**88.96**	81.29	80.11

Table 2. Average recall (%) at top 1% (@1%) and top 1 (@1) for each of the models trained on Oxford RobotCar, U.S. and R.A..

5.4. Results visualization

In addition to quantitative results, we select and show qualitative results of some correctly retrieved matches in Fig. 7. A full traversal is chosen randomly as the reference map on four benchmark datasets, respectively. Then we choose four query point clouds from other randomly selected traversals on their respective datasets, with each representing one sample submap from individual testing areas. For each instance, the query point cloud and the top 3 retrieved matches are shown on the left. It becomes clear that the best match has a very similar scene as the query point cloud. Besides, we display the location of each point cloud in the reference map on the right. For each query, the location of the top 1 result (indicated by the blue circle) is correctly overlapped with the query location (represented by the red cross). It shows that the proposed network indeed has the ability to recognize places.

6. Discussion

6.1. Ablation study

Ablation studies evaluate the effectiveness of different proposed components in our network, including both the PointOE module and self-attention unit. We also analyze the performance of the proposed HPHN quadruplet loss. All experiments are conducted on Oxford RobotCar.

PointOE module and self-attention unit. We test the effectiveness of the proposed PointOE module and the self-attention unit, using PointNetVLAD and PCAN as baselines (PN_VLAD, PCAN). We first just integrate either PointOE module or self-attention unit into PointNetVLAD, referred as PN_VLAD-OE and PN_VLAD-S. We then combine both two components into PointNetVLAD, denoted as PN_VLAD-SOE. Besides, we replace PointNet by PointNet++ [32] in the local descriptor extraction stage, referred to as PN++_VLAD. All networks are trained with lazy quadruplet loss, with results shown in Table 3.

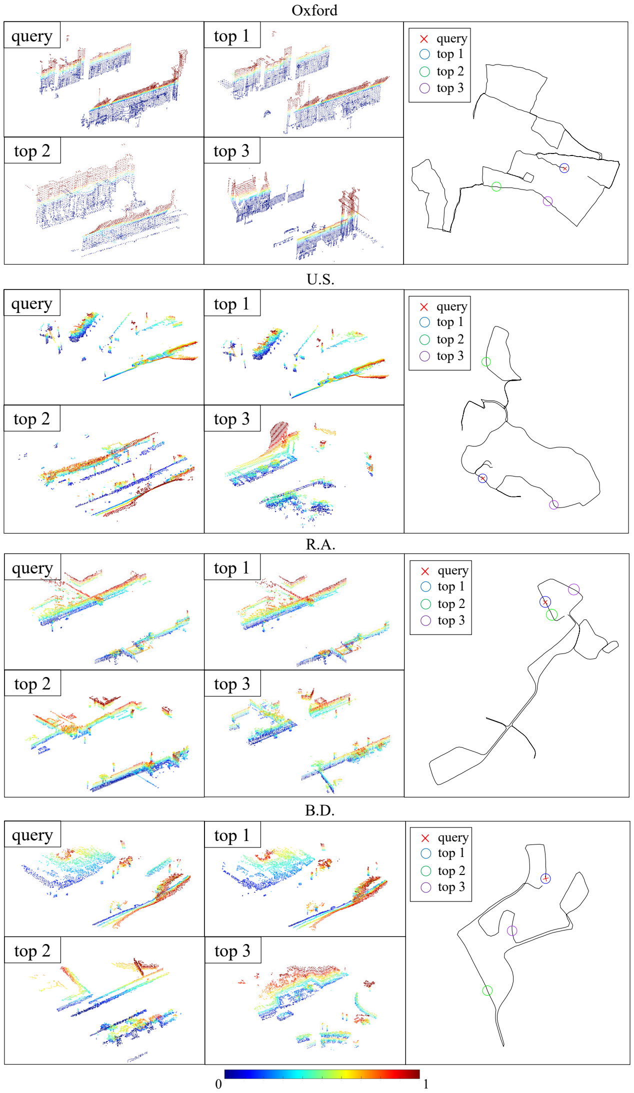


Figure 7. Visualizations of example retrieval results of SOE-Net on four benchmark datasets. For each retrieval, the query point cloud and the top 3 retrieved results are displayed. Locations of these point cloud is also indicated in the associated reference map. Colors in these point clouds represents heights above the ground.

Comparing with PointNetVLAD, PN_VLAD-S sees an improvement of 5.7% and 10.27% on the average recall at top 1% and top 1, respectively. The performance of PN_VLAD-S also exceeds the recall of PCAN by 2.9% and 3.98%, respectively, indicating the proposed self-attention unit is more effective than the attention strategy used in PCAN. This is due to the context information has a significant effect on aggregating local descriptors into a global one, and our self-attention unit can learn long-range spatial relationships between local descriptors. With the proposed PointOE module, our model brings significant improvements on the average recall by 11.19% and 19.45%, respectively, when compared with PointNetVLAD. Besides, PointNet++ enhances PointNet features with a hierarchical encoding pipeline, but still does not explicitly encode orientation. The comparison with PN++_VLAD demonstrates the superiority of OE for 3D descriptor learning for place recognition. Combining both modules can improve the performance by 12.40% and 21.44% on average recall, respectively. The ablation studies demonstrate the significant role of each module in SOE-Net.

	Ave recall @1%	Ave recall @1
PN_VLAD	81.01	62.76
PN++_VLAD	89.10	76.23
PCAN	83.81	69.05
PN_VALD-S	86.71	73.03
PN_VALD-OE	92.20	82.21
PN_VALD-SOE	**93.41**	**84.20**

Table 3. Ablation studies of self-attetnion unit and PointOE module on Oxford RobotCar. The results show the average recall (%) at top 1% (@1%) and at top 1 (@1) for each model.

HPHN quadruplet loss. To evaluate the proposed HPHN quadruplet loss, we compare the performance of the proposed SOE-Net trained with different losses. As shown in Table 4, the network performance is better when trained on the proposed HPHN quadruplet loss. The performance on Oxford RobotCar reaches 96.40% recall at top 1% and 89.47% recall at top 1, exceeding the same model trained with the lazy quadruplet loss by 2.99% and 5.17%, respectively, demonstrating the superiority of the proposed HPHN quadruplet loss.

	Ave recall @1%	Ave recall @1
Lazy quadruplet	93.41	84.20
HPHN quadruplet	**96.40**	**89.37**

Table 4. Results of the average recall (%) at top 1% and at top 1 of SOE-Net trained with different losses on Oxford RobotCar.

	SOE-Net			DAGC		
	D=128	D=256	D=512	D=128	D=256	D=512
Oxford	95.30	96.40	**96.70**	84.43	**87.49**	85.72
U.S.	91.24	93.17	**94.47**	81.17	**83.49**	83.02
R.A.	90.53	**91.47**	91.00	72.39	**75.68**	74.46
B.D.	85.88	88.45	**89.29**	69.57	**71.21**	68.74

Table 5. Results of the average recall (%) at top 1% of different global descriptor dimensions on Oxford RobotCar. D is the output dimension of global descriptors.

6.2. Output dimension analysis

In this section, we analyze the performance of the global descriptor with different output dimensions. The results of average recall at top 1% for the global descriptor produced by SOE-Net and DAGC are shown in Table 5. We can draw two conclusions from this table: (1) our method outperform DAGC, even if the generated global descriptor has a smaller dimension; (2) when the output dimension decreases from 256 to 128, the performance of SOE-Net only declines by around 1%-3% on each benchmark. When the dimension expands to 512, the performance only changes by about 0.3%-1%. This implies the robustness of our method against different output dimensions.

6.3. Values of margin analysis

In this section, we explore the network performance with different margins in the HPHN quadruplet loss using Oxford RobotCar. Table 6 shows results of average recall at top 1% and top 1 with different margins for the SOE-Net architecture. Seen from the table, SOE-Net achieves the best performance with a margin value of 0.5. When the values expand to 0.7, the performance steadily degrades. This implies the distance between positive and negative pairs is sufficient with lower values of margin. On the other hand, when the value is set to 0.4, the performance decreases. So, we set the fixed value of margin as 0.5 in our network.

Margin	Ave recall @1%	Ave recall @1
0.4	95.87	88.84
0.5	**96.40**	**89.37**
0.6	96.23	89.30
0.7	95.63	88.46

Table 6. Margin analysis in the HPHN quadruplet loss. We choose SOE-Net as a baseline and evaluate it on Oxford RobotCar.

7. Conclusion

In this paper, we propose a novel end-to-end network SOE-Net for point cloud based retrieval. We design a PointOE module and a self-attention unit, using information from neighboring points and long-range context dependency to enhance the feature representation ability. In addition, we propose a novel HPHN quadruplet loss that achieves more discriminative and generalizable global descriptors. Experiments show that our SOE-Net improves the retrieval performance over state-of-the-art methods significantly. According to discussions on experimental results, especially ablation studies, we can discover that PointOE module contributes most to the performance of the SOE-Net. There is also one notable limitation of the SOE-Net, which regards that the margin in the HPHN quadruplet loss needs to be set beforehand. In the future, we will explore adaptive margins that can better distinguish positive and negative pairs.

Acknowledgement We sincerely thank Max Hoedel for proofreading. This research was supported by the China Scholarship Council.

References

[1] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[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] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2017.

[5] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8958–8966, 2019.

[6] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[7] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[8] Juan Du, Rui Wang, and Daniel Cremers. Dh3d: Deep hierarchical 3d descriptors for robust large-scale 6dof relocalization. *arXiv preprint arXiv:2007.09217*, 2020.

[9] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2017.

[10] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *European conference on computer vision*, pages 224–237. Springer, 2004.

[11] Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. Texture mapping for 3d reconstruction with rgb-d sensor. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4645–4653, 2018.

[12] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[13] 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.

[14] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013.

[15] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018.

[16] 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.

[17] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 153–161, 2017.

[18] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *European conference on computer vision*, pages 791–804. Springer, 2010.

[19] Haomin Liu, Guofeng Zhang, and Hujun Bao. Robust keyframe-based monocular slam for augmented reality. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–10. IEEE, 2016.

[20] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2372–2381, 2017.

[21] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2570–2579, 2019.

[22] 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.

[23] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui 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*, pages 2831–2840, 2019.

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[25] Fan Lu, Guang Chen, Yinlong Liu, Zhongnan Qu, and Vittorio Murino. Rskdd-net: Random sample-based keypoint detector and descriptor. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12–21, 2019.

[27] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan.

Aslfeat: Learning local features of accurate shape and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[29] 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.

[30] Jane Palmer, Brenda Boardman, Agneta Persson, HM Suvilehto, W Herbert, and S Loerx. Delight: domestic efficient lighting. *Environmental Change Unit, University of Oxford, United Kingdom*, 1998.

[31] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017.

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[33] Timo Röhling, Jennifer Mack, and Dirk Schulz. A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 736–741. IEEE, 2015.

[34] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[35] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE, 2008.

[36] 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 *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 224–232, 2020.

[37] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conference on Computer Vision*, pages 630–646. Springer, 2018.

[38] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[39] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[40] Wenxiao Zhang and Chunxia Xiao. Pcan: 3d attention map learning using contextual information for point cloud based retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12436–12445, 2019.