RINet: Efficient 3D Lidar-Based Place Recognition Using Rotation Invariant Neural Network

Lin Li, *Graduate Student Member, IEEE*, Xin Kong, *Member, IEEE*, Xiangrui Zhao, *Graduate Student Member, IEEE*, Tianxin Huang, Wanlong Li, Feng Wen, *Member, IEEE*, Hongbo Zhang, *Member, IEEE*, and Yong Liu

***Abstract*—LiDAR-based place recognition (LPR) is one of the basic capabilities of robots, which can retrieve scenes from maps and identify previously visited locations based on 3D point clouds. As robots often pass the same place from different views, LPR methods are supposed to be robust to rotation, which is lacking in most current learning-based approaches. In this letter, we propose a rotation invariant neural network structure that can detect reverse loop closures even training data is all in the same direction. Specifically, we design a novel rotation equivariant global descriptor, which combines semantic and geometric features to improve description ability. Then a rotation invariant siamese neural network is implemented to predict the similarity of descriptor pairs. Our network is lightweight and can operate more than 8000 FPS on an i7-9700 CPU. Exhaustive evaluations and robustness tests on the KITTI, KITTI-360, and NCLT datasets show that our approach can work stably in various scenarios and achieve state-of-the-art performance.**

***Index Terms*—Recognition, localization, semantic scene understanding.**

I. Introduction

PLACE recognition is a fundamental issue in computer vision and robotics, which is also known as loop closure in SLAM (Simultaneous Localization and Mapping). Robots typically need to identify the current location by comparing the query sensor data with a historical map database. It helps correct cumulative errors of odometry and estimate positions in GPS denied situations, e.g., under overhanging trees or near high buildings. Camera and LiDAR are the two most commonly used sensors in SLAM. Visual place recognition (VPR) has been studied extensively and continuously [1], [2], as image data is easy to acquire and contains rich textures. However, image-based methods are quite sensitive to illumination and seasonal changes, limiting their application in large-scale scenes. Whilst, LiDAR captures accurate geometric structures and is rarely affected by appearance changes, which has recently attracted widespread attention.

Most LPR methods [3]–[5] are geometric-based by encoding geometric features of point clouds into the scene representation. Recently, some studies [6]–[8] show that semantics help boost performance, especially robustness. However, semantic-based LPR methods are few, and how to effectively use semantics is still an open issue.

LiDARs typically have a 360-degree field of view, and robots often return with different viewpoints. Thus, the LPR algorithms are required to identify the same place with a large orientation difference [9], or are supposed to be rotation invariant. Some methods use brute force search [4], [8], [10] or transform to the frequency domain [5] to achieve rotation invariance. However, many learning-based methods do not consider viewpoint changes or expect rotation robustness via simple data augmentation by randomly rotating training samples. When the input data rotates, the output of these neural networks also changes, producing unlike descriptors for the same scene. Therefore, neural networks for LPR need to be specially designed to achieve rotation invariance.

In this paper, we propose a structurally rotation invariant neural network, which can focus more on learning the scene's characteristics instead of being entangled in the orientation of point clouds. To achieve this goal, we first design a novel rotation equivariant global descriptor, leveraging semantic and geometric features for better discriminative ability. Then we improve the typical convolutional and pooling layer to strictly ensure our neural network rotation invariant, which is further formed as a siamese network for similarity prediction. Our main contribution is summarized as follows:

- We propose a novel rotation equivariant global descriptor for LPR, which combines semantic and geometric information for better description ability.
- We propose a novel rotation invariant neural network, which achieves strict rotation invariance structurally.
- The inference speed of our network can reach more than 8000 FPS on an i7-9700 CPU and 20,000 FPS on an

Manuscript received September 8, 2021; accepted January 24, 2022. Date of publication February 11, 2022; date of current version February 24, 2022. This letter was recommended for publication by Associate Editor Zetao Chen and Editor Cesar Cadena Lerma upon evaluation of the reviewers' comments. The work of Xin Kong was supported by the China Scholarship Council-Imperial Scholarship . *(Corresponding author: Yong Liu.)*

Lin Li, Xiangrui Zhao, and Tianxin Huang are with the State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310000, China (e-mail: 22032043@zju.edu.cn; xiangruizhao@zju.edu.cn; 21725129@zju.edu.cn).

Xin Kong is with the Department of Computing, Imperial College London, SW72AZ London, U.K. (e-mail: xinkong@zju.edu.cn).

Wanlong Li, Feng Wen, and Hongbo Zhang are with the Huawei Noah's Ark Lab, Beijing 100095, China (e-mail: liwanlong@huawei.com; wenfeng3@huawei.com; zg_hongbo@sina.com).

Yong Liu is with the State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310000, China, and also with the Huzhou Institute of Zhejiang University, Hangzhou 310027, China (e-mail: yongliu@iipc.zju.edu.cn).

Our code will be available at: https://github.com/lilin-hitcrt/RINet.

Digital Object Identifier 10.1109/LRA.2022.3150499

NVIDIA GeForce GTX 1080 Ti GPU, making it applicable for mobile platforms with limited resources.

- Exhaustive experiments on the KITTI [11], KITTI-360 [12], and NCLT [13] datasets demonstrate the state-of-the-art performance and robustness of our approach. Our code will be publicly available.

II. Related Work

A. 3D LiDAR-Based Place Recognition

As LiDAR can obtain rich and accurate environmental 3D geometric information, most LPR methods focus on extracting statistical features of geometric distributions. Rusu *et al.* [14] propose a global descriptor called VFH for 3D point clouds, which encodes geometry and viewpoint simultaneously. Steder *et al.* [15] represent point clouds as range images and then use the feature point-based method for place recognition. Boss and Zlot [16] propose a keypoint voting approach to speed up feature matching. Röhling *et al.* [17] encode the height of the point as a histogram to get the global statistics of the scenes. M2DP [3] projects the point cloud to multiple 2D planes and then uses the statistical information on each plane to generate descriptors. Scan context [4] divides the point cloud into different blocks according to the radius and azimuth direction, and uses reserves maximum heights for each block. LiDAR-Iris [5] proposes a LiDAR-Iris image representation and transforms it into the frequency domain to achieve pose-invariant loop closure detection. Seed [18] encodes the topological relation of the segmented objects as the global descriptor, achieving rotation invariant and insensitive to translation variation. Recently, some methods obtained good results by leveraging intensity information [10], [19], [20].

Benefit by powerful neural networks, learning-based methods show great potential in LPR. OREOS [21] projects point clouds onto 2D range images and extract global descriptors by CNN. SegMatch [22] and SegMap [23] extract 3D features of the segmented objects and then perform place recognition via feature matching. Locus [24] aggregates the features extracted by SegMatch into a global descriptor. This method can encode the topological and temporal information of the scenes. PointNetVLAD [25] uses PointNet [26] to extract local features and uses NetVLAD [1] to aggregate global features. DiSCO [9] uses an encoder-decoder to extract features from the scan context image representation, which is further converted to the frequency domain to eliminate the effect of rotation. NDT-Transformer [27] converts the point cloud into an NDT representation and adopts the Transformer [28] to extract the global descriptor. FusionVLAD [29] fuse features extracted from the spherical-view and top-down view to generate more robust descriptors.

Some recent studies explore the usage of semantics for scene representation. SGPR [6] and GOSMatch [30] represent point clouds as semantic graphs. OverlapNet [7] combines various information such as semantics, normals to achieve a full description of the scene. SSC [8] uses semantics to improve the performance of scan context. In this paper, we leverage semantics and geometric features to build a discriminative rotation equivariant global descriptor.

B. Rotation Invariant Neural Network

Though deep learning has made breakthroughs in point cloud processing recently, typical neural networks [26], [31], [32] are not robust to rotation. Thus, making network rotation invariant has aroused researchers' concern. Kim *et al.* [33] propose a local-to-global representation algorithm, which improves robustness to rotation. SE(3)-Transformers [34] propose a variant of the self-attention module for 3D point clouds and graphs, which is equivariant under continuous 3D roto-translations. PRIN [35] proposes the Spherical Voxel Convolution and Point Resampling to extract rotation invariant features for each point. Esteves *et al.* [36] model 3D data with multi-valued spherical functions and propose a spherical convolutional network to learn SO(3) Equivariant Representations. However, the above methods are designed generally for 3D point clouds, which is not specialised for LiDAR data and either the LPR task. In fact, the rotation of autonomous robots only occurs in the yaw direction, which can simplify the problem. Thus, we propose a rotation invariant neural network suitable for LiDAR data and LPR.

III. Methodology

The proposed approach mainly consists of three parts: descriptor generation, feature extraction and similarity prediction, as shown in Fig. 1. Given a pair of point clouds, we first convert them into rotation equivariant global descriptors. Then for the obtained global descriptors, we use a rotation invariant siamese neural network to extract features further. After feature extraction, each point cloud is encoded as a fixed-length vector. Finally, we compare the feature vectors to get the similarity score between the input point clouds.

A. Preliminaries

Rotation equivariant: Let a point cloud be $P \in R^{N \times 3}$, where N is the number of points. We define our descriptor as $D \in R^{N_s \times N_l}$, where N_s and N_l are the length and the number of channels, respectively. Our descriptors can be processed efficiently by 1-D convolution. We denote the operation of encoding the point cloud as $\mathbb{G} : R^{N\times 3} \rightarrow R^{N_s \times N_l}$. If the rotation $\mathbb{R}$ on P is equivariant to the shift $\mathbb{S}$ on D, we call $\mathbb{G}$ is rotation equivariant:

$$\mathbb{S}_{[\rho\theta]}(D) = \mathbb{S}_{[\rho\theta]}(\mathbb{G}(P)) = \mathbb{G}(\mathbb{R}_\theta(P)) \tag{1}$$

where $\theta \in [0, 2\pi)$ is the rotation angle, ρ is a scale factor, and $[*]$ is the rounding function. Let V be the input of a specific layer in the network. If any operation $\mathbb{F}$ is shift equivariant [37], the following formula holds:

$$\mathbb{S}_{[\epsilon d]}(\mathbb{F}(V)) = \mathbb{F}(\mathbb{S}_d(V)) \tag{2}$$

where d and ϵ are the amount of shift and a scale factor, respectively. Since the rotation on the point cloud is equivalent to the shift on the descriptor in our method, we also call the above property rotation equivariant. For convenience, we will refer to the shift as 'rotation' hereafter.

Rotation invariant: We say that the operation $\mathbb{F}$ is rotation invariant [38] if the following formula holds:

$$\mathbb{F}(V) = \mathbb{F}(\mathbb{S}_d(V)) \tag{3}$$

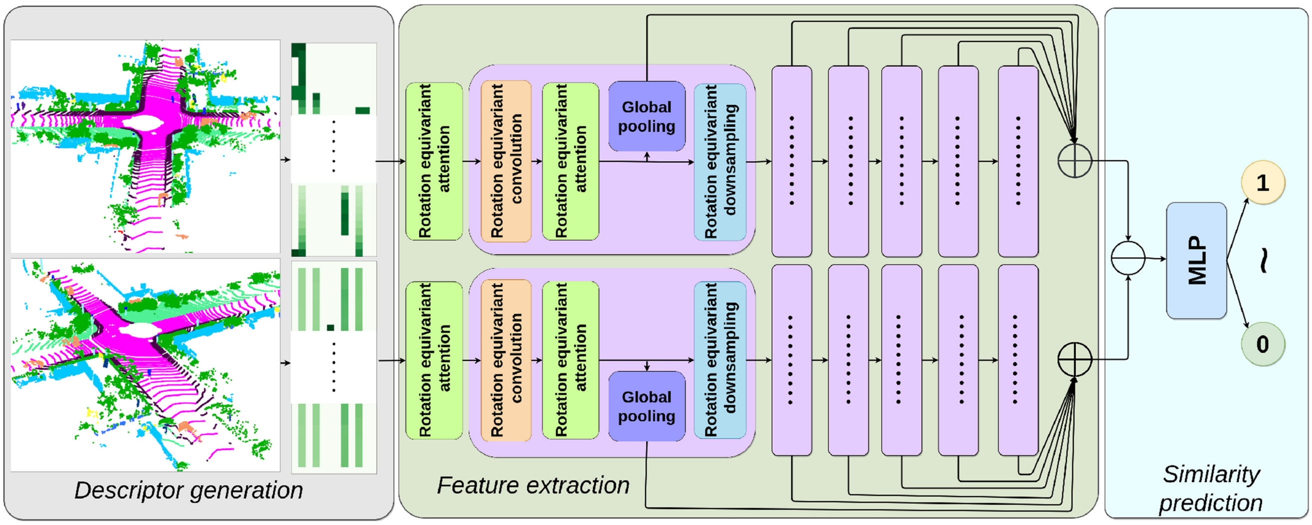

Descriptor generation
Rotation equivariant attention
Rotation equivariant convolution
Rotation equivariant attention
Global pooling
Rotation equivariant downsampling
Feature extraction
MLP
1
~
0
Similarity prediction

Fig. 1. The pipeline of the proposed approach mainly consists of three parts: descriptor generation, feature extraction, and similarity prediction. In descriptor generation, we combine semantic and geometric info to encode the scene as a rotation equivariant descriptor. In feature extraction, we use a rotation invariant siamese network to further extract features from the descriptor and finally obtain a feature vector of length N_f. We use an MLP to score the similarity between feature vectors.

Further, we define $\mathbb{F}$ as sum-rotation-invariant if the following formula holds:

$$\sum \mathbb{F}(V) = \sum \mathbb{F}(\mathbb{S}_d(V)) \tag{4}$$

where the summation is over the entries of $\mathbb{F}(V)$ and $\mathbb{F}(\mathbb{S}_d(V))$. It's easy to prove that a rotation equivariant operation (2) must be sum-rotation-invariant [38]. Therefore, as long as each operation in our network is rotation equivariant, we can achieve rotation invariant by adding a global pooling operation at the end.

B. Rotation Equivariant Global Descriptor

Given a point cloud, we first convert it to the polar coordinate $P = \{(r_i, \varphi_i, l_i)\}$, where r_i is the polar radius, φ_i is the polar angle, and l_i is the semantic label. Then we divide the point cloud into N_s sectors according to the polar angle. Each sector is defined by:

$$S_i = \left\{ (r,l) \,\middle|\, \frac{i \times 2\pi}{N_s} - \pi \leq \varphi < \frac{(i+1) \times 2\pi}{N_s} - \pi \right\} \tag{5}$$

where $i \in [0, N_s - 1]$, $\forall (r, \varphi, l) \in P$. For each sector S_i, we further divide it into N_l channels according to semantics:

$$B_{ij} = \{r | l = L(j), \forall (r,l) \in S_i\} \tag{6}$$

where $i \in [0, N_s - 1]$, $j \in [0, N_l - 1]$, and L is the set of N_l semantic labels. Finally, we obtain descriptor $D \in R^{N_s \times N_l}$:

$$D(i,j) = \mathbb{E}(B_{ij}) = \min_{r \in B_{ij}} (r) \tag{7}$$

Function $\mathbb{E}$ encodes B_{ij} as a scalar. We define $\mathbb{E}$ as taking the minimum distance ('road' is special, taking the maximum distance) to get the contour of the scene (Fig. 2(b)). Since the laser of the traditional LiDAR cannot penetrate objects, the idea of selecting the closest point is very intuitive. Notably, due to discretization errors, our descriptors are not strictly rotation equivariant, but this is harmless (proved in Section IV-B). Fig. 2 demonstrates the process of generating descriptors.

C. Rotation Invariant Neural Network

Rotation equivariant convolution: CNNs have always been considered shift-equivariant (or rotation equivariant in our case). However, some recent studies point out that stride is the key reason why CNNs are not strictly shift equivariant. However, restricting stride to 1 cannot achieve complete rotation equivariant due to the influence of the boundary. In point clouds, 0 and 359 degrees are adjacent, but they are at the head and tail in descriptors. An intuitive solution is to use circular convolution [39] with a stride of 1.

We take the first layer of our network to explain how the rotation equivariant convolution works. Suppose the input signal is $V \in R^{N_s \times N_l}$ and the out put is $\overline{V}$, the rotation equivariant convolution is formulated as:

$$\begin{aligned} \overline{V}(i) &= (V * K)(i) \\ &= \sum_{c=0}^{N_l - 1} \sum_{m=-M}^{M} V((i-m) \mod N_s, c) K(M+m, c) \end{aligned} \tag{8}$$

where $i \in [0, N_s - 1]$, and $K \in R^{(2M+1) \times N_l}$ is the kernel. However, the above rotation equivariant convolution cannot downsample V. We need to design additional rotation equivariant downsampling operations.

Rotation equivariant downsampling: Downsampling can increase the receptive field, aggregate features, and reduce the subsequent calculation. However, the typical max pooling and average pooling are not rotation equivariant. Using anti-aliasing-based downsampling [37] that blurs the feature map before downsampling, can alleviate this problem. But recent studies report that nonlinear functions will affect the effect of anti-aliasing [38]. To achieve strict rotation equivariant, we adopt the adaptive polyphase sampling (APS) method proposed in [38].

Suppose we need to downsample $V \in R^{N_s \times N_l}$ to $\overline{V} \in R^{\frac{N_s}{k} \times N_l}$ (N_s must be divisible by k). We first divide V into k sub-vectors:

$$V_i = \{V(j) | (N_s \bmod j) \equiv i, j \in [0, N_s - 1]\} \tag{9}$$

where $i \in [0, k-1]$. Then downsampled $\overline{V}$ is defined as:

$$\overline{V} = \operatorname*{argmax}_{V_i} \|V_i\|_1 \tag{10}$$

where $\|\cdot\|_1$ is L1 norm. Thus, rotation equivariance is strictly ensured by taking sub-vector with the maximum L1 norm.

Rotation equivariant attention: As depicted in Section III-B that each channel of the descriptor corresponds to different semantic objects. Intuitively, different semantic objects have different contributions to the whole scene representation. For example, for humans, buildings are more distinguishable than roads. Therefore, we propose the rotation equivariant attention module, which gives different weights to each channel to enable our network to pay more attention to those representative semantic objects. The following formula is used to calculate the weight for each channel of V:

$$C = \text{Sigmoid}\left(\left(\frac{1}{N_s}\sum_{i=1}^{N_s} V(i)\right) W + b\right) \tag{11}$$

where $W \in R^{N_l \times N_l}, b \in \mathbb{R}^{1 \times N_l}$ are learnable weight matrix and bias vector, respectively. Since we average all the entries of V, the output weight C is rotation invariant. Then we use C to weight V to obtain the module's output $\overline{V}$:

$$\overline{V} = V \cdot C^T \tag{12}$$

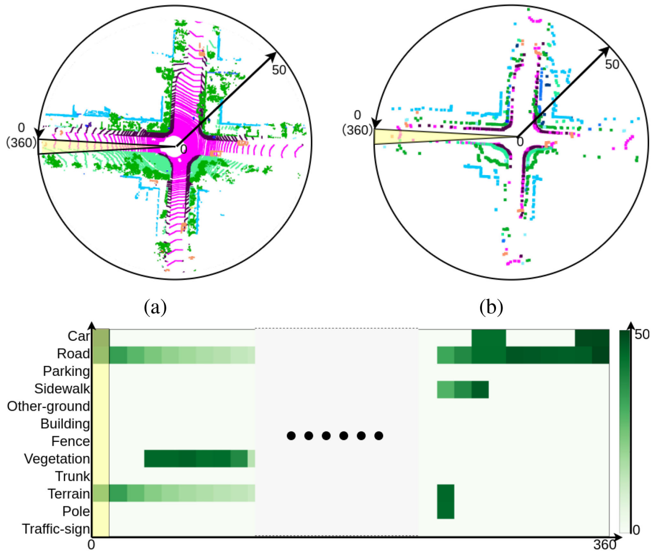
It is easy to know that this module is rotation equivariant.

Feature extraction: As shown in the middle of Fig. 1, our feature extraction network has two branches that share weights. Each branch is formed by stacking sub-networks with the same structure composed of several basic rotation equivariant modules as introduced above. In each sub-network, we use the global average pooling to obtain the rotation invariant feature vector. Finally, we concatenate these feature vectors from different sub-networks to form a global rotation invariant feature vector of length N_f.

Similarity prediction: As shown in the right of Fig. 1, feature vectors f_1 and f_2 are obtained after feature extraction, and their similarity is calculated as:

$$\text{score} = \text{Sigmoid}(\text{MLP}(|f_1 - f_2|)) \tag{13}$$

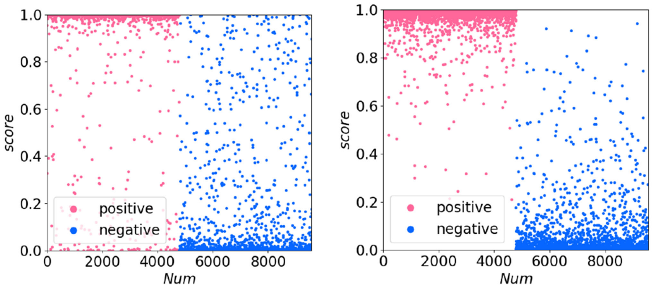
where $|\cdot|$ denotes absolute value. Thus the similarity *score* $\in [0, 1]$ is invariant to the order of f_1 and f_2. The MLP consists of two linear layers (288×128, 128×1) and a Leaky ReLU layer in between.



(a) (b)

(c)

Fig. 2. We first convert (a) The semantic point cloud to the polar coordinate. Then we divide it into N_s sectors according to the polar angle. For each sector, we split it into N_l channels based on semantics. In each channel, we keep the points (b) With the smallest polar radius. Finally, we get (c) The descriptor $D \in R^{N_s \times N_l}$, each entry representing the polar radius of the corresponding point.



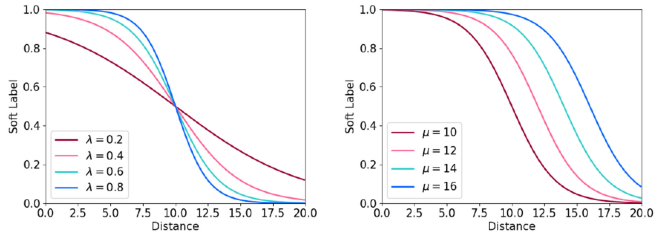
(a) Binary cross-entropy. (b) Soft binary cross-entropy.

Fig. 3. There are more extreme values in the predictions (a) When training the network with the BCE loss. (b) Using the SBCE loss makes distribution more desirable.

SGPR [6] treats place recognition as a classification problem and uses binary cross-entropy (BCE) loss to train the model. However, we find that the model trained in this way tends to output extreme scores. As shown in Fig. 3(a), the network will give very high scores (close to 1) to some negative samples. This will cause the recall rate of the network at 100% accuracy to be close to 0, which will seriously affect the performance. We suppose that the network is more inclined to predict extreme scores (very close to 0 or 1) as it is only given ground-truth labels with a score of 0 or 1 during training. To solve this problem, we adopt the soft binary-cross entropy (SBCE) loss, where the labels are continuously distributed between 0 and 1. In our case, the sample label is defined by the following formula:

$$\text{label} = 1 - \frac{1}{1 + e^{\lambda(\mu - \delta)}} \tag{14}$$

where δ is the distance between the sample pairs. Fig. 4 shows the influence of λ and μ on labels. As shown in Fig. 3(b), the network trained with SBCE give a more reasonable score distribution (no extreme scores are given when the classification is wrong) than BCE.



(a) Fix μ. (b) Fix λ.

Fig. 4. Figure (a) Shows how the Equ.14 curve changes with the value of λ while Figure (b) With μ.

IV. Experiments

A. Dataset and Implementation Details

We evaluate on KITTI, KITTI-360, and NCLT datasets.

KITTI: The KITTI dataset contains 11 sequences (00 to 10) with ground-truth poses, which is collected by a 64-beam LiDAR (Velodyne HDL-64E) in Karlsruhe. Among all sequences, only 00, 02, 05, 06, 07, 08 contain loop closures, and only 08 has reverse loop closures. For semantics, we adopt the ground truth labels from SemanticKITTI dataset [40] (Ours-SK), which provides 28 categories. Following previous work [6], our descriptors choose 12 of them (car, road, parking, sidewalk, other-ground, building, fence, vegetation, trunk, terrain, pole, traffic-sign). To test the robustness of our approach to noisy semantic predictions which is common in real systems, we report the results of using the predictions from Cylinder3D [41] (Ours-CY).

Following SGPR [6] training strategy, we adopt k-fold cross-validation, where each sequence is considered as a fold. In the testing phase, we follow the testing set in SSC [8] and ensure all methods are evaluated on the same test data.

KITTI-360: The KITTI-360 and KITTI datasets are collected in different places of the same city. KITTI-360 has 6 sequences with loop closures (0000, 0002, 0004, 0005, 0006, 0009) and is more challenging than KITTI due to the larger number of reverse loop closures in each sequence. KITTI-360 provides semantic labels for the whole map instead of each single point cloud frame. So we use the semantic labels provided by SSC open-sourced code, which are processed from the ground-truth labels. In addition, we also evaluate with semantic predictions provided by Cylinder3D.

Notably, we directly evaluate on KITTI-360 with the model only trained on KITTI dataset.

NCLT: The NCLT dataset is collected at the University of Michigan, which contains 27 sequences over 14 months. Different from the KITTI and KITTI-360 datasets, NCLT is a long-term dataset and equips a 32-beam LiDAR. The NCLT dataset does not label semantics for the point clouds, and it is difficult to perform transfer learning due to the large distribution gap. To overcome the lack of semantics, we modify our descriptor, by replacing the 12 semantic channels of the original descriptor with dividing the point cloud into 12 parts by height (Ours-HE). This descriptor is also used in ablation study Section IV-D.

For NCLT dataset, we follow the data splitting in DiSCO [9], which splits each run into two disjoint parts for training and testing. We use 6 runs (2012-03-17, 2012-05-26, 2012-06-15, 2012-08-20, 2012-09-28, 2012-10-28) for evaluation and the rest runs to train our model. If the distance between the query and retrieved scan is less than 1.5 m, the localization is considered successful.

When training our model on KITTI and NCLT, every two point clouds in the training sequences form a sample pair, and we use the method in Section III-C to calculate their labels. We set μ and λ in Eq. 14 to 10 and $\frac{2}{3}$, respectively. In Section III-B, we set $N_s = 360$ and $N_l = 12$. Our feature extraction network has a total of 6 layers and produces a feature vector of length $N_f = 288$. Our code is based on PyTorch using Adam optimizer with learning rate 0.02.

B. Place Recognition Performance

KITTI and KITTI-360: The precision-recall curve and the maximum value of the F_1 scores are used to measure the performance. The F_1 score is defined as follow, where P is precision and R is recall.

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{15}$$

We compare the proposed approach with the state-of-the-art methods, including four non-learning methods (M2DP [3], Scan Context [4] (SC), LiDAR Iris [5] (LI), and Semantic Scan Context [8] (SSC)) and four learning-based methods (PointNetVLAD [25] (PV), SGPR [6], Locus [24], and DiSCO [9]). To verify the generalization ability of our method, we use the model trained on the KITTI dataset when testing on the KITTI-360 dataset.

As shown in Fig. 5 and Table I, our method performs well on KITTI, surpassing other methods in the indicator of mean F_1 max score with a large margin. The results on sequence 08 are intriguing, as the loop closures in the training set are all in the same direction, while sequence 08 only contains reverse loop closures. Thanks to the strictly rotation invariant design, our approach can still achieve good results on sequence 08 even only trained on the loop closures in the same direction. The effect of rotation/viewpoint changes is further proved in Section IV-C. Note that our method only trained on KITTI achieves amazing results even by directly testing on KITTI-360, showing the strong generalization ability. The results of Ours-CY are worse than that of Ours-SK as expected, but it is still very competitive, which shows that our algorithm is robust to semantic predictions containing noises. As we do not retrain the models when testing Ours-CY but directly use the models from Ours-SK, this further indicates our method's strong generalization performance.

To visually show the effect of the attention module, we randomly select 12 samples and draw the weights in Fig. 6. The attention module assigns higher weights to 'traffic-signs' and 'other-ground'. On the contrary, it assigns very low weights to 'road' and 'sidewalk'. This verifies our assumption in Section III-C that our network can learn to weight different semantic objects.

NCLT: The Average Recall@1 (AR@1) and Average Recall@1% (AR@1%) are used as metrics. We compare with five advanced methods: Locus [24], PointNetVLAD [25] (PV), Scan Context [4] (SC), OREOS [21], and DiSCO [9]. Note that semantics are not available on NCLT, so we use geometric-based descriptors instead. As shown in Table III, our method achieves competitive results, which proves that our method is robust to seasonal changes and can work well on different data. We can expect that our method will perform better if semantics

TABLE I
F1 MAX SCORES ON KITTI AND KITTI-360 DATASETS

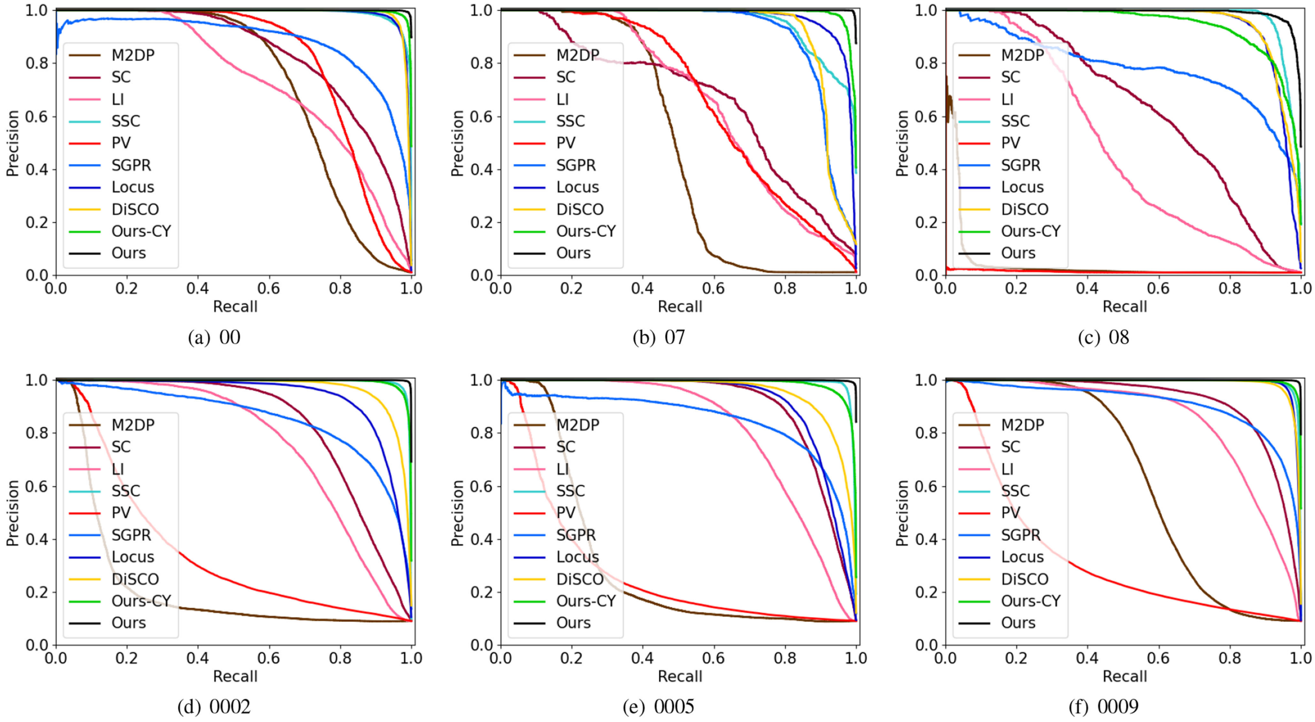
	KITTI							KITTI-360						
	00	02	05	06	07	08	Mean	0000	0002	0004	0005	0006	0009	Mean
M2DP [3]	0.708	0.717	0.602	0.787	0.560	0.073	0.575	0.423	0.209	0.246	0.311	0.397	0.620	0.368
SC [4]	0.750	0.782	0.895	0.968	0.662	0.607	0.777	0.831	0.771	0.811	0.843	0.834	0.851	0.824
LI [5]	0.668	0.762	0.768	0.913	0.629	0.478	0.703	0.688	0.704	0.714	0.747	0.720	0.782	0.726
SSC [8]	0.951	0.891	0.951	0.985	0.875	0.940	0.932	0.921	0.974	0.975	0.974	0.978	0.970	0.965
PV [25]	0.779	0.727	0.541	0.852	0.631	0.037	0.595	0.352	0.349	0.325	0.285	0.295	0.330	0.323
SGPR [6]	0.820	0.751	0.751	0.655	0.868	0.750	0.766	0.818	0.788	0.795	0.798	0.833	0.843	0.813
Locus [24]	0.957	0.745	**0.968**	0.948	0.921	0.900	0.907	0.908	0.871	0.896	0.858	0.878	0.966	0.896
DiSCO [9]	0.964	0.892	0.964	0.990	0.897	0.903	0.935	0.922	0.916	0.932	0.890	0.909	0.957	0.921
Ours-CY	0.978	0.947	0.917	0.978	0.967	0.869	0.943	0.935	0.966	0.968	0.949	0.951	0.976	0.958
Ours-SK	**0.992**	**0.948**	0.954	**1.000**	**0.995**	**0.959**	**0.975**	**0.956**	**0.993**	**0.991**	**0.991**	**0.993**	**0.993**	**0.986**

All models are trained only on the KITTI dataset. The best scores are marked in bold and the second best scores are underlined.

TABLE II
ROBUSTNESS TEST

	Occlusion								Rotation							
	00	02	05	06	07	08	Mean	Cmp	00	02	05	06	07	08	Mean	Cmp
M2DP [3]	0.199	0.138	0.283	0.140	0.113	0.046	0.153	-0.422	0.276	0.282	0.341	0.316	0.204	0.201	0.270	-0.305
SC [4]	0.724	0.751	0.845	0.904	0.616	0.552	0.732	-0.045	0.719	0.734	0.844	0.898	0.606	0.546	0.725	-0.052
LI [5]	0.627	0.710	0.679	0.859	0.585	0.383	0.641	-0.062	0.667	0.764	0.772	0.912	0.633	0.470	0.703	0.000
SSC [8]	0.919	0.881	**0.929**	0.948	0.847	**0.911**	0.906	**-0.026**	0.955	0.889	0.952	0.986	0.876	0.943	0.934	**+0.002**
PV [25]	0.547	0.570	0.295	0.589	0.444	0.031	0.413	-0.182	0.083	0.090	0.490	0.094	0.064	0.086	0.151	-0.444
SGPR [6]	0.649	0.604	0.619	0.542	0.625	0.531	0.595	-0.171	0.772	0.716	0.723	0.640	0.748	0.678	0.713	-0.053
Locus [24]	0.915	0.719	0.919	0.880	0.865	0.824	0.854	-0.053	0.944	0.726	**0.960**	0.927	0.911	0.877	0.891	-0.016
DiSCO [9]	0.894	0.845	0.891	0.938	0.846	0.832	0.874	-0.061	0.960	0.891	0.952	0.985	0.894	0.892	0.929	-0.006
Ours-SK	**0.971**	**0.916**	0.881	**0.983**	**0.970**	0.909	**0.938**	-0.037	**0.992**	**0.942**	0.954	**1.000**	**0.990**	**0.962**	**0.973**	-0.002
Ours-SK*	-	-	-	-	-	-	-	-	0.992	0.948	0.954	1.000	0.995	0.959	0.975	0.000

F_1 max scores on KITTI dataset when the point cloud are randomly occluded 30° FoV and rotated around z-axis.
Cmp is the comparison with the standard results shown in Table I.



(a) 00 (b) 07 (c) 08

(d) 0002 (e) 0005 (f) 0009

Fig. 5. Precision-Recall curves on KITTI and KITTI-360 datasets. Figure (a)-(c) shows the results on the KITTI dataset. Figure (d)-(f) shows the results on the KITTI-360 dataset. The models used in all experiments are only trained on KITTI.

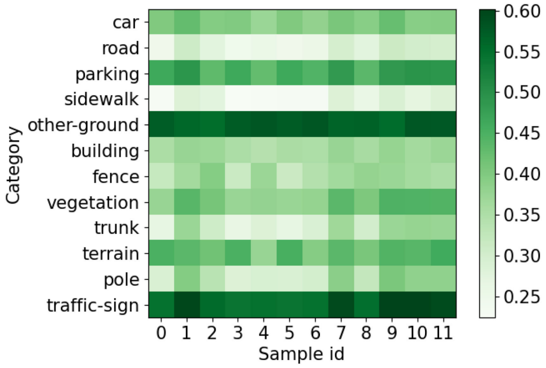


Fig. 6. We randomly select 12 samples and draw the weights calculated by the first attention module. The result reveals that our network focuses on representative semantic objects.

TABLE III
AR@1 AND AR@1% ON NCLT DATASET

Approach	Locus [24]	PV [25]	SC [4]	OREOS [21]	DiSCO [9]	Ours-HE
AR@1	34.32	44.10	63.58	58.02	**89.08**	87.64
AR@1%	53.39	76.27	92.35	91.44	99.14	**99.25**

are available. The effect of semantics is further illustrated in Section IV-D.

C. Robustness Test

We follow the experiments in SGPR [6] to test the robustness of the proposed method. The following robustness test experiments are all based on the KITTI dataset.

Occlusion: In the case of long-term and large-scale localization, the occlusion of the scene is inevitable. To simulate this situation, we randomly remove the points in a range of 30 degrees in the yaw direction of each point cloud. As shown in Table II, our method still surpasses other methods in most sequences. Our method has the least performance degradation among all learning-based methods, and perform better than most non-learning methods except SSC. This experiment proves that our method is robust to occlusion.

Viewpoint Changes: Viewpoint change is another problem that often arises in large-scale localization. We randomly rotate the point cloud to simulate this situation. We design two experiments to prove that only the step of generating descriptors in our method is not strictly rotation equivariant. The first experiment (Ours-SK) is the same as the other methods, randomly rotating the point cloud. The second experiment (Ours-SK*) randomly shifts the descriptors intead of rotating the point cloud. As shown in Table II, Ours-SK are slightly inconsistent with the original while Ours-SK* is strictly consistent with the original. It proves that our network is strictly rotation invariant, while our descriptor is approximately rotation equivariant due to the discretization error during projection. Our method is more robust to rotation among all learning-based methods. DiSCO [9] transforms the feature map to the frequency domain to achieve rotation invariance. However, due to the use of ordinary convolutional neural networks before FFT, their network is not strictly rotation invariant. Whilst, due to the influence of discretization, their descriptors are not strictly rotation equivariant either.

TABLE IV
CONTRIBUTION OF INDIVIDUAL COMPONENTS

RIConv	RIPool	RIAtten	Semantic	Mean	Cmp
	√	√	√	0.879	-0.096
√		√	√	0.940	-0.035
√	√		√	0.958	-0.017
√	√	√		0.923	-0.052
√	√	√	√	0.975	-

TABLE V
TIME COST ON KITTI 08

Methods	Batch Size	GPU	Description	Retrieval	Total
SGPR [6]	4071	9367	0.239|4071	0.022|201394	0.261
Locus [24]	1	9963	4459.971|4071	2.898|201394	4462.869
DiSCO [9]	512	7453	4.040|4071	0.270|201394	4.310
Ours	4071	**2188**	**0.200**|4071	**0.011**|201394	**0.211**
Ours-CPU	4071	-	0.495|4071	0.131|201394	0.626

The time unit is in second and GPU consumption is in MB.

D. Ablation Study

To specify the individual contribution of each component, we design the ablation study on the KITTI dataset. In the experiment, we randomly shift the descriptors to explore the influence of each component on the rotation invariance of the network. To explore the effects of rotation equivariant convolution (RIConv) and rotation equivariant downsampling (RIPool), we replace them with typical convolution (the same kernel size and stride as our RIConv) and average pooling (the same downsampling rate as our RIPool), respectively. To verify the effects of the rotation equivariant attention module (RIAtten), we directly remove it from the network. We use the geometry-based descriptor proposed for NCLT (Ours-HE Section IV-A) to explore the contribution of semantics. As shown in Table IV, the average F_1 max scores decrease by 0.096 and 0.035, respectively, when using typical convolution and pooling. It is easy to know that typical convolution and pooling operations will destroy the rotation invariance of the network. The average F_1 max score drops 0.017 when the rotation equivariant attention module is removed. As demonstrated in Section IV-B, the attention module enables the network to pay more attention to representative semantic objects. When using height-based descriptors (Ours-HE), the average F_1 max score is reduced by 0.052, showing that semantics can help generate more representative descriptors.

E. Efficiency

We report the efficiency of learning-based methods on KITTI's sequence 08. All experiments are conducted on the same computer with an Intel Core i7-9700 and an NVIDIA GeForce GTX 1080 Ti GPU. For each method, we count the time required for extracting features from all the sequence 08 frames (Description) and scoring all candidate pairs (Retrieval). We test our method on GPU (Ours) and CPU (Ours-CPU) separately, while all other methods are tested on the GPU. As shown in Table V, our method is faster than other methods in all stages. Our method takes a total of 0.211 s to complete the evaluation of KITTI's sequence 08, of which it takes 0.200 s to extract the features from 4071 descriptors (about 20,000 FPS) and 0.011 s to evaluate the similarities of 201394 feature pairs. On the CPU, our method takes 0.626 s to complete the entire experiment, of which 0.495 s (about 8000 FPS) is used to extract features, and 0.131 s is used to calculate similarity. In addition, our method requires very little GPU memory. When the batch size is 4071, only 2188 M GPU memory is required. The above results indicate that our method is promising in mobile robot platforms with constrained resources.

V. Conclusion

This paper proposes RINet, an efficient network for 3D Lidar-Based Place Recognition, which is structurally robust to viewpoint changes. Specifically, a novel rotation equivariant global descriptor is firstly proposed by combining semantic and geometric features, and then a lightweight siamese network is applied, including three basic rotation invariant modules: rotation equivariant convolution, downsampling and attention. Benefiting from the structurally rotation invariant design, the proposed approach achieves advanced performance on the KITTI, KITTI-360, and NCLT datasets while operating very efficiently.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[2] S. Garg and M. Milford, “SeqNet: Learning descriptors for sequence-based hierarchical place recognition,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4305–4312, Jul. 2021.

[3] L. He, X. Wang, and H. Zhang, “M2DP: A novel 3D point cloud descriptor and its application in loop closure detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 231–237.

[4] G. Kim and A. Kim, “Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4802–4809.

[5] Y. Wang, Z. Sun, C. Z. Xu, S. E. Sarma, J. Yang, and H. Kong, “LiDAR iris for loop-closure detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5769–5775.

[6] X. Kong *et al.*, “Semantic graph based place recognition for 3D point clouds,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8216–8223.

[7] X. Chen *et al.*, “OverlapNet: Loop closing for LiDAR-based SLAM,” in *Proc. Robot., Sci. Syst.*, 2020.

[8] L. Li *et al.*, “SSC: Semantic scan context for large-scale place recognition,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2092–2099.

[9] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, “DiSCO: Differentiable scan context with orientation,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2791–2798, Apr. 2021.

[10] H. Wang, C. Wang, and L. Xie, “Intensity scan context: Coding intensity and geometry relations for loop closure detection,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2095–2101.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[12] Y. Liao, J. Xie, and A. Geiger, “KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D,” 2021, *arXiv:2109.13410*.

[13] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, “University of michigan north campus long-term vision and LiDAR dataset,” *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2015.

[14] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, “Fast 3D recognition and pose using the viewpoint feature histogram,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2155–2162.

[15] B. Steder, G. Grisetti, and W. Burgard, “Robust place recognition for 3D range data based on point features,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 1400–1405.

[16] M. Bosse and R. Zlot, “Place recognition using keypoint voting in large 3D LiDAR datasets,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 2677–2684.

[17] T. Röhling, J. Mack, and D. Schulz, “A fast histogram-based similarity measure for detecting loop closures in 3-D LiDAR data,” in*Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 736–741.

[18] Y. Fan, Y. He, and U.-X. Tan, “Seed: A segmentation-based egocentric 3D point cloud descriptor for loop closure detection,” in *Proc. IEEE/RSJ Int. conf. Intell. Robots Syst.*, 2020, pp. 5158–5163.

[19] J. Guo, P. V. K. Borges, C. Park, and A. Gawel, “Local descriptor for robust place recognition using LiDAR intensity,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1470–1477, Apr. 2019.

[20] K. P. Cop, P. V. K. Borges, and R. Dubé, “Delight: An efficient descriptor for global localisation using LiDAR intensities,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3653–3660.

[21] L. Schaupp, M. Bürki, R. Dubé, R. Siegwart, and C. Cadena, “OREOS: Oriented recognition of 3D point clouds in outdoor scenarios,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 3255–3261.

[22] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, “SegMatch: Segment based place recognition in 3D point clouds,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5266–5272.

[23] R. Dubé *et al.*, “SegMap: Segment-based mapping and localization using data-driven descriptors,” *Int. J. Robot. Res.*, vol. 39, no. 2-3, pp. 339–355, 2020.

[24] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, and C. Fookes, “Locus: LiDAR-based place recognition using spatiotemporal higher-order pooling,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5075–5081.

[25] M. A. Uy and G. H. Lee, “PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4470–4479.

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[27] Z. Zhou *et al.*, “NDT-transformer: Large-scale 3D point cloud localisation using the normal distribution transform representation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5654–5660.

[28] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[29] P. Yin, L. Xu, J. Zhang, and H. Choset, “FusionVLAD: A multi-view deep fusion networks for viewpoint-free 3D place recognition,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2304–2310, Apr. 2021.

[30] Y. Zhu, Y. Ma, L. Chen, C. Liu, M. Ye, and L. Li, “GosMatch: Graph-of-semantics matching for detecting loop closures in 3D LiDAR data,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5151–5157.

[31] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet : Deep hierarchical feature learning on point sets in a metric space,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5100–5109.

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

[33] S. Kim, J. Park, and B. Han, “Rotation-invariant local-to-global representation learning for 3D point cloud,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 8174–8185.

[34] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling, “SE(3)-transformers: 3D roto-translation equivariant attention networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1970–1981.

[35] Y. You *et al.*, “Pointwise rotation-invariant network with adaptive sampling and 3D spherical voxel convolution,” in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 12717–12724.

[36] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, “Learning SO(3) equivariant representations with spherical CNNs,” *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 588–600, 2020.

[37] R. Zhang, “Making convolutional networks shift-invariant again,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 12712–127722.

[38] A. Chaman and I. Dokmanic, “Truly shift-invariant convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3773–3783.

[39] S. Schubert, P. Neubert, J. Pöschmann, and P. Protzel, “Circular convolutional neural networks for panoramic images and laser data,” in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 653–660.

[40] J. Behley *et al.*, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[41] X. Zhu *et al.*, “Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9939–9948.