SE(3)-Equivariant Point Cloud-based Place Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract: This paper reports on a new 3D point cloud-based place recognition framework that uses SE(3)-equivariant networks to learn SE(3)-invariant global descriptors. We discover that, unlike existing methods, learned SE(3)-invariant global descriptors are more robust to matching inaccuracy and failure in severe rotation and translation configurations. Mobile robots undergo arbitrary rotational and translational movements. The SE(3)-invariant property ensures the learned descriptors are robust to the rotation and translation changes in the robot pose and can represent the intrinsic geometric information of the scene. Furthermore, we have discovered that the attention module aids in the enhancement of performance while allowing significant downsampling. We evaluate the performance of the proposed framework on real-world data sets. The experimental results show that the proposed framework outperforms state-of-the-art baselines in various metrics, leading to a reliable point cloud-based place recognition network.

Keywords: Place Recognition, SE(3)-Invariant, SE(3)-Equivariant Representation Learning, 3D Point Clouds

1 Introduction

Place recognition can be defined as linking the sensor's in-situ observations and the prebuilt reference map. Among numerous 2D (RGB, thermal, and event-triggered) and 3D (stereo, LiDAR, and RGB-D) sensors [1], 3D sensors are gaining popularity and have recently been extensively researched. Modern service robots, autonomous cars [2], and drones [3] are widely equipped with consumer-level 3D sensors due to their better environment perception ability and decreasing prices. Thus, place recognition techniques with 3D data can be used in estimating the agent's location in scenarios such as self-driving vehicles, autonomous indoor navigation, or scientific exploration. Place recognition, also known as loop closure detection, is a critical component in Simultaneous Localization and Mapping (SLAM). It enables a robot to determine if it has seen a place before and provides loop closure candidates [4]. With a correct loop closure, the SLAM system can eliminate accumulated drift from the odometry and improve the mapping accuracy [5].

Extracting consistent features from 3D data is an important research topic but remains underexplored and unsolved [6]. One key issue in present place recognition methods is that they do not consider transformation changes in data or expect robustness via simple data augmentation [7]. Take data measured on vehicles as an example. If the vehicle changes lanes, though it is still in the same location, translation differences exist in the data. Furthermore, if it travels to an intersection where the previous pose is in a different direction, then rotation changes exist in the data. Since existing works do not consider these transformation changes, their performance is sensitive to transformation variations in the training and testing point cloud samples. As such, the extracted global descriptors change substantially when the point clouds are rotated or translated, resulting in place recognition failure as descriptors are matched incorrectly. Our research aims at designing a rotation and translation-invariant global descriptor for point clouds, called SE(3)-invariant feature, to solve the transformation-sensitivity problem.

The attention mechanism of transformers [8] enables networks to learn the correlation between input features and obtain the importance of each feature. Some place recognition frameworks like

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

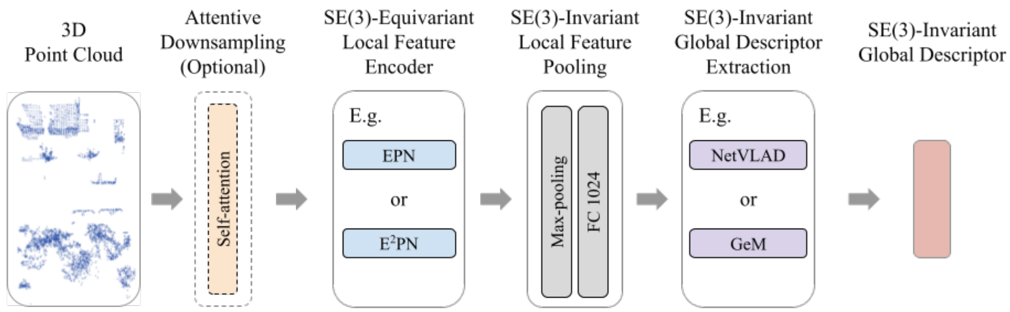


Figure 1: Overview of the proposed SE(3)-equivariant point cloud-based place recognition pipeline. Optionally, 3D point clouds are preprocessed with an attentive downsampling process. Next, SE(3)-equivariant local features are learned using SE(3)-equivariant networks. In this work, we use EPN and E^2PN. Then, SE(3)-invariant local features are extracted by max-pooling. Lastly, SE(3)-invariant global descriptors are computed by global pooling methods. In this work, we use NetVLAD and Generalized Mean (GeM). The global descriptors are SE(3)-invariant and can perform place recognition tasks.

PCAN [9] use an attention mechanism on local features to re-weight each feature. However, they usually apply an attention mechanism to feature space to learn the importance of each feature. In our work, we apply the attention mechanism on the 3D points to learn which point to reserve during the downsampling process.

In this paper, we propose a place recognition framework that exploits SE(3)-invariant features to perform place recognition in challenging rotation and translation scenes (Figure 1). We propose to learn SE(3)-equivariant local features via a group-equivariant encoder; in this work, we use a modified Equivariant Point Network (EPN) [10] and its more efficient variant E^2PN [11]. However, our pipeline is agnostic to the particular approach for learning equivariant features. Then, SE(3)-invariant global descriptors are learned by aggregating local features using NetVLAD [12] or Generalized Mean (GeM) [13]. Moreover, we apply a self-attention mechanism for downsampling point clouds before the equivariant encoder block to decrease memory usage and increase efficiency in training. We train our network on Oxford RobotCar Dataset [14] and evaluate on Oxford, in-house [15], and KITTI odometry benchmark [16]. We also validate our proposed framework for rotation and translation scenes. Experimental results show that our approach consistently outperforms existing state-of-the-art approaches. The experiment of the trained network on unseen data sets verifies the generalizability and scalability of our proposed framework.

The main contributions of this work can be summarized as follows:

1. We propose a new pipeline for place recognition using SE(3)-equivariant encoders to learn SE(3)-invariant descriptors with only geometric information from 3D point clouds. The proposed method is robust against arbitrary rotation and translation of robot poses. It is generalizable and scalable to unseen data sets, thereby removing the need for data augmentation.
2. We apply a self-attention mechanism to downsample point clouds which maintains place recognition in high performance up to 50 % downsampling rate.
3. The code is open-sourced and will be publicly available after receiving the final decision.

2 Related Work

3D point clouds generated by LiDAR (light detection and ranging), stereo cameras, RGB-D cameras, or other sensors obtain rich and accurate environmental 3D geometric information. We first review place recognition works that utilize the geometric information from 3D point clouds in section 2.1. Then, we present existing works that utilize the point cloud descriptors with rotation-invariant or translation-invariant properties in section 2.2. Later, we discuss existing group-equivariant networks for 3D point cloud in section 2.3. In section 2.4, we provide brief introduction on designing attention mechanisms to improve place recognition performance.

2.1 Geometry-based Place Recognition

Previously, place recognition using point clouds relied on histograms or hand-engineered features such as Fast Histogram [17], M2DP [18] and Scan Context [19]. M2DP [18] projects 3D point clouds to multiple 2D planes and constructs global descriptors from singular vectors of density signatures. Scan Context [19] represents the point cloud in the polar axis and encodes the height of the observed points into the representation.

PointNetVLAD [15] is a pioneering work to apply a learning-based feature extractor to place recognition tasks. It combines PointNet [20] and NetVLAD [12] to allow end-to-end representation training from a given 3D point cloud. LPD-Net [21], which proposes adaptive local feature extraction and graph-based neighborhood aggregation to construct a global descriptor. OverlapNet [22] constructs range images to learn the overlap score and the yaw angle between two inputs. MinkLoc3D [23] presents sparse voxelized point cloud representation and sparse 3D convolutions. LCDNet [24] provides an estimated pose in addition to representation learning in the network. However, these algorithms are not robust to rotational and translational pose changes.

2.2 Exploiting Symmetry in Place Recognition

Considering that the observer may be in different orientations or locations is critical, researchers propose some hand-crafted, rotation-invariant features to perform place recognition more robustly and accurately. Yin et al. [25] propose a heading-invariant feature that uses histograms of range in a LiDAR scan ring to deal with the change of heading angle of the vehicle. Scan Context [19] uses ring keys, the occupancy ratio of rings in scan context, as rotation-invariant features. It is further generalized in their later work Scan Context++ [26] to include lateral invariance by augmentation based on urban road assumption. FreSCo [27] uses frequency-domain Scan Context to perform place recognition with translation and rotation invariance. LiDAR-Iris [28] encodes height information into eight-bit binary code and uses Fourier transform to estimate the translation between two LiDAR-Iris images to remove the rotational difference between LiDAR scans. Xu et al. [29] build polar grid height coding image descriptor which is rotationally invariant. While these hand-crafted features are rotation-invariant, some structural information is ignored when composing them. Later, deep learning features are widely used since their performances surpass hand-crafted features [30].

Only a few works try to encode rotation-invariant or translation-invariant features into learning-based place recognition algorithms. PointNetVLAD [15] and LCDNet [24] try to increase robustness by randomly rotating input point clouds during training. RINet [7] exploits additional semantic information and combines it with rotation equivariant convolution to achieve rotation-invariant. OverlapNet [22] and OverlapTransformer [31] use range images to make the feature yaw-angle-invariant. Lu et al. [32] propose a RING descriptor that is translation-invariant after the discrete Fourier transform procedure and is yaw-angle-invariant. SeqSphereVLAD [33] uses spherical convolution to extract orientation-invariant descriptors from point clouds in spherical view. RPR-Net [34] constructs rotation-invariant feature using rotation-invariant convolution. Nevertheless, these strategies do not consider both 3D rotation and translation differences in the pose, thus might not be sufficient in more challenging scenarios.

2.3 Group-Equivariant Networks for 3D Point Clouds

While only a small number of works take rotation-equivariant and translation-equivariant into consideration in place recognition tasks, a series of works design network architectures with the equivariance property for general feature learning. Esteves et al. [35] propose Spherical CNNs (Convolutional Neural Networks), which map 3D models into spherical functions and use spherical convolutions to generate equivariant feature maps. Vector Neuron [36] proposes a SO(3)-equivariant network that replaces scalars with 3-vectors in the neurons. Equivariant Point Network (EPN) [10], which we adopt in our work, performs SE(3) separable convolution, which separates 6D convolution into convolutions in the 3D Euclidean space and in SO(3). It enables SE(3)-equivariant feature learning in a computationally affordable way. E^2PN [11] proposes a lightweight variant of SE(3)-equivariant network for point clouds, which we also test in our work. These networks generally address the rotation-equivariant feature learning problem in classification and segmentation tasks. They are only tested with point clouds in single object shapes but have not been tested much on 3D point clouds in real-world outdoor scenes. We propose a new pipeline for place recognition that exploits symmetry via group-equivariant networks. This work is the first attempt to develop SE(3)-equivariant place recognition framework to bridge the gap between the group-equivariant and place recognition literature.

2.4 Attention Mechanism in Place Recognition

An attention mechanism has been applied to some place recognition tasks to utilize the neighborhood context better. PCAN [9] predicts the significance of each local feature using an attention mechanism. Similarly, SOE-Net [37] includes this technique to learn the contextual features. Retriever [38] builds an attention mechanism between local features and a latent code to construct global descriptors. OverlapTransformer [31] includes Transformer to learn spatial relations of different features before feeding into NetVLAD. Among these applications, attention mechanisms are used to learn the importance of the local features. In this work, we explore applying the attention mechanism to the input 3D points to learn the points' significance.

3 Methodology

This section details our proposed EPN-NetVLAD framework for SE(3)-invariant place recognition using 3D point clouds. Figure 1 presents an overview of the proposed approach. The framework consists of three parts: attentive downsampling, SE(3)-invariant local feature extraction, and SE(3)-invariant global descriptor generation. We will fully discuss each component in the following subsections.

3.1 Attentive Downsampling

3D point clouds measured from LiDAR or RGB-D sensors may contain hundreds of thousands of points. To perform place recognition efficiently in neural networks, we exploit the attention mechanism to downsample point cloud measurements while preserving meaningful information. For a point cloud with N points $P \in \mathbb{R}^{N\times 3}$, we apply the multi-head attention module [8] using the PyTorch library [39] to learn the attention weights $W_{atten} \in \mathbb{R}^{N\times 3}$ from the input point cloud. Multi-head attention is defined as $\mathrm{MultiHead}(X) = \mathrm{Concat}(head_1, head_2, head_3)W^O$, where $\mathrm{Concat}(\cdot)$ does the features concatenation. The number of parallel attention heads is set as 3. Each attention head is defined as $head_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$. Here, we set query $Q = XW_i^Q$, key $K = XW_i^K$, and value $V = XW_i^V$, where, X is the input point cloud P to perform self-attention and learn correlation between the input 3D points.

With the attention weights, we summarize over the feature space to obtain point-wise attention weights $W_{pw-atten} \in \mathbb{R}^N$, which represent the significance of each point. $W_{pw-atten} = \sum_{i=1,2,3} W^i_{atten}$, where $W^i_{atten} \in \mathbb{R}^N$ is the attention weight in dimension i. We select top-k attention weights and keep the corresponding points $P' = \mathbb{R}^{k\times 3}$.

3.2 Local SE(3)-Equivariant Features

Learning equivariant representation from point clouds can provide efficiency and generalizability in challenging robot perception tasks. *Equivariance* is a form of symmetry for functions that preserve the transformation applied on the input to the output.

Equivariance generalizes the concept of *invariance*, which means that the output of functions is independent of the transformations applied to the input. Mathematically, a function $f_{inv} : X \to X$ is *invariant* to a set of transformations T, if for any $t \in T$, $f_{inv}(x) = f_{inv}(t \cdot x), \forall x \in X$.

The general linear group of degree n, denoted by $\mathrm{GL}_n(\mathbb{R})$, is the set of all $n \times n$ nonsingular real matrices, where the group binary operation is the ordinary matrix multiplication. The three-dimensional (3D) special orthogonal group, denoted by $\mathrm{SO}(3) = \{R \in \mathrm{GL}_3(\mathbb{R}) \mid RR^\mathsf{T} = I_3, \det R = +1\}$ is the rotation group on $\mathbb{R}^3$, where I_3 denotes the 3×3 identity matrix. The 3D special Euclidean group, denoted by

$$\mathrm{SE}(3) = \{H = (R, t) \mid R \in \mathrm{SO}(3), t \in \mathbb{R}^3\}$$

is the group of rigid transformations, i.e., direct isometries on $\mathbb{R}^3$ [40].

In this work, we leverage Equivariant Point Network (EPN) [10] and E^{2}PN [11] to learn the SE(3)-equivariant feature and capture the inherent symmetry of 3D point cloud data. In the original EPN [10], given a 3D point x, a rotation g, a feature representation function $\mathcal{F} : \mathbb{R}^3 \times \mathrm{SO(3)} \rightarrow \mathbb{R}^D$, and a kernel $h : \mathbb{R}^3 \times \mathrm{SO(3)} \rightarrow \mathbb{R}^D$, the discretized SE(3)-equivariant convolutional operator is defined as the dot product between the translated and rotated kernel and the function $\mathcal{F}$:

$$(\mathcal{F} * h)(x, g) = \sum_{x_i \in \mathcal{P}} \sum_{g_j \in G} \mathcal{F}(x_i, g_j) h(g^{-1}(x - x_i), g_j^{-1} g), \tag{1}$$

where P and G are the discretized sets corresponding to $\mathbb{R}^3$ and $\mathrm{SO}(3)$, respectively. To reduce the computation cost in 6D convolution, the authors separate the kernel h into two smaller kernels representing SE(3) point convolution and SE(3) group convolution, respectively. This design preserves SE(3)-equivariant features from the input point cloud while maintaining affordable computation.

We also experimented with E^{2}PN [11], which is a lightweight and more efficient variant of EPN [10]. E^{2}PN leverages quotient representations to embed SO(3)-equivariance in a spherical feature space, resulting in much fewer feature dimensions than EPN. Therefore, it drastically reduces memory consumption and runtime while preserving the rotational equivariance. Such property is highly relevant to our task since we work with large-scale point clouds in an outdoor environment.

3.3 Local SE(3)-Invariant Feature Pooling

After learning SE(3)-equivariant features $f_e(P)$, pooling is then applied to extract SE(3)-invariant features. To avoid the group attentive pooling failing if the point cloud is circularly symmetric as discussed in [10], we propose to apply max-pooling on the rotational dimension for each spatial point to generate SE(3)-invariant features and increase the robustness for different shapes of point clouds. SE(3)-equivariant features represent as $f_e(P) \in \mathbb{R}^{N \times C \times R}$, where P is the input point cloud, $f_e(\cdot)$ is the mapping from point cloud to SE(3)-equivariant features, N is number of points, C is number of local features, and R is the number of rotation group discretization. In the max-pooling step, we only keep the maximum feature from one of the R discretized rotation groups. After max-pooling, the SE(3)-invariant feature is then represent as $f_{inv}(P) \in \mathbb{R}^{N \times C}$. The last part of the local feature extractor is a linear layer to map the SE(3)-invariant features to the desired dimension. See Figure 1 for an illustration.

3.4 Global SE(3)-Invariant Place Representation

Global descriptors are computed by aggregating local features using NetVLAD or Generalized Mean (GeM) [13]. NetVLAD learns cluster centers of VLAD (Vector of Locally Aggregated Descriptors) in a CNN framework. The output descriptors V are adopted for describing the places and are given in (2). This equation shows j-th dimensions of the i-th descriptor, where x is the local feature. w_k, b_k, and c_k are trainable parameters to learn the center of cluster k.

$$V(j, k) = \sum_{i=1}^{N} \frac{e^{w_k^\intercal x_i + b_k}}{\sum_{k'} e^{w_{k'}^\intercal x_i + b_{k'}}} (x_i(j) - c_k(j)). \tag{2}$$

GeM is a trainable pooling layer that generalizes max-pooing and mean-pooling. With local feature input x, the output of GeM pooling is defined in (3). Where p is a pooling parameter that can be set manually. When $p \rightarrow \infty$, the process is max-pooling. When $p = 1$, it is mean-pooling.

$$f_{GeM} = (\frac{1}{|x|} \sum_{x_i \in x} x_i^p)^{\frac{1}{p}}. \tag{3}$$

To learn discriminative and generalizable global descriptors for performing place recognition tasks, we use lazy quadruplet loss proposed by Uy and Lee [15]. For each iteration of training, there are an anchor point cloud P_a, a "positive" point cloud P_p that is similar to the anchor point cloud, and some "negative" point clouds $\{P_n\}$ that are dissimilar to the anchor point cloud, and a random point cloud in the training set P_{n^*}. The lazy quadruplet loss defined in (4) can minimize the L2 distance between anchor and positive representation $\delta_p = d(f(P_a), f(P_p))$ while maximizing the distance between anchor and some negative representation $\delta_{n_j} = d(f(P_a), f(P_{n_j})), P_{n_j} \in \{P_n\}$. α and β are constant values to provide margin.

$$Loss(P_a, P_p, P_n, P_{n^*}) = \max_j([\alpha + \delta_p \delta_{n_j}]_+) + \max_k([\beta + \delta_p \delta_{n_k^*}]_+). \quad (4)$$

4 Experimental Results and Discussion

We construct SE(3)-invariant place recognition descriptors using the described method. In this section, we examine the performance of place recognition, SE(3)-invariant properties, and the design of attentive downsampling.

4.1 Model Training

We train our networks on Oxford RobotCar [14] benchmark created by Uy and Lee [15]. Oxford benchmark contains 45 sequences of a vehicle taking measurements using SICK LMS-151 2D LiDAR in similar routes for different times, days, and seasons. Each point cloud is a submap of a pre-built map. The ground points are removed, and the point clouds are normalized to be zero mean and inside the range of [-1, 1]. Training and testing sets are geographically split with a ratio of 70 % and 30 %. For creating training tuples, a ground truth location within 10 meters is considered a positive pair, while a location larger than 50 meters is considered a negative sample. We train with 21,711 sub-maps. We trained and tested our method on a system equipped with Intel i9-10900K CPU with a 3.7 GHz processor and an Nvidia GeForce RTX 3090.

In EPN-NetVLAD, Point clouds are downsampled to 2048 points using the attention mechanism. We construct EPN-NetVLAD with two layers of EPN, one with 32 local features and one with 64 local features. In EPN, we set the number of discretized rotation groups R as 60. EPN is followed with max-pooling and a linear layer to map local features to 1024 dimensions. Then, we use NetVLAD to learn global descriptors with dimensions of 256. The network is trained for 30 epochs with a learning rate of 5×10^{-5}. Each training tuple consists of one query point cloud, one "positive" point cloud, one "negative" point cloud, and another random point cloud. The hyper-parameters in lazy quadruplet loss in set as $\alpha = 0.5, \beta = 0.2$. The network parameters are optimized by ADAM [41].

We construct E^{2}PN-NetVLAD with two layers of E^{2}PN, one with 32 local features and one with 64 local features. The number of discrete rotation groups in E^{2}PN is 12. Then, it follows the same setting for NetVLAD as in EPN-NetVLAD. For GeM in E^{2}PN-GeM, we follow MinkLoc3D's structure and set pooling parameter $p = 3$.

Note that we do not need random rotation during the training process since the network is designed to generate the same descriptor as we rotate or translate the point cloud. The decreased need for data augmentation is an advantage of the proposed framework.

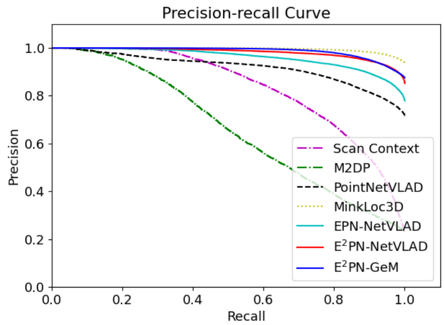
4.2 Place Recognition Evaluation

In place recognition tasks, precision and recall are the two well-established evaluation metric [42]. Precision is the percentage of true loop closures among all the places we recognize. Recall is the percentage of places we recognize among all true loop closures. The definition is shown in (5), where TP is the number of true-positive cases, FP represents the number of false-positive cases, and FN stands for the number of false-positive cases. The F1 score is introduced and defined in the same equation to obtain a balancing metric between precision and recall.

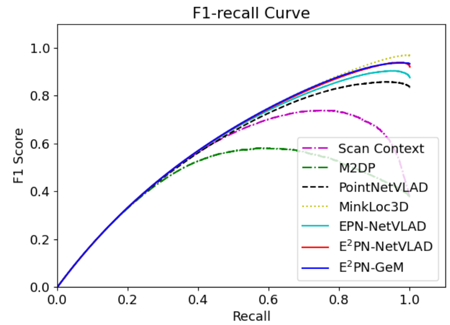
$$\text{precision} = \frac{TP}{TP + FP}; \text{ recall} = \frac{TP}{TP + FN}; \text{ F1} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (5)$$

4.2.1 Oxford and in-house benchmark

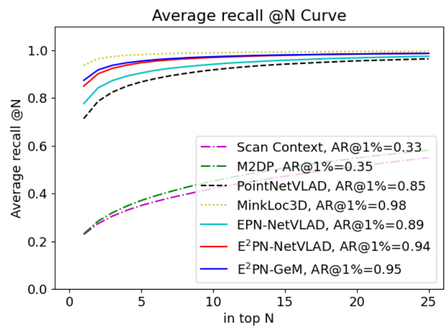
We first evaluate the performance of the proposed method on the Oxford benchmark. The Oxford RoboCar Dataset consists of data collected by vehicles driving in a similar route at different times and seasons. Hence, every sequence revisits the path traveled by other sequences. When



(a)



(b)



(c)

Figure 2: Experimental results of proposed methods (E^{2}PN-GeM in blue line, E^{2}PN-NetVLAD in red line, and EPN-NetVLAD in cyan line), state-of-the-art approaches MinkLoc3D [23], PointNetVLAD [15], M2DP [18], and Scan Context [19] on Oxford benchmark.

performing the evaluation, we generate the SE(3)-invariant global descriptor for each input point cloud. Then, we find the top 1, top 25, and top 1% of candidates' matches similar to the query point cloud in each sequence. We calculate the precision, recall rate, and average among different query point clouds in different sequences. The average recall curve represents the model performance for the top 25 matches. With these evaluation metrics and scikit-learn library [43], we report precision-recall curves, F1-recall curves, and average recall curves of the proposed method and other state-of-the-art methods are shown in Figure 2. EPN-NetVLAD, E^{2}PN-NetVLAD, E^{2}PN-GeM, PointNetVLAD [15], and MinkLoc3D [23] are trained on the same Oxford benchmark training set. However, MinkLoc3D is trained with a more efficient training strategy. Scan Context [19] and M2DP [18] construct hand-engineered features to perform place recognition. The figure shows that the proposed network E^{2}PN-NetVLAD and EPN-NetVLAD outperform PointNetVLAD, which shares the same global feature extraction method. MinkLoc3D and E^{2}PN-GeM both use GeM pooling for global feature extraction. Though MinkLoc3D performs the best among all methods, E^{2}PN-GeM and E^{2}PN-NetVLAD still perform consistently within 5% of difference.

To show the generalizability of the proposed method, we also evaluate all methods on in-house data sets with three kinds of regions that are unseen to the network, including the university sector (U.S.), residential area (R.A.), and business district (B.D.). In-house data sets are generated by Uy and Lee [15] and are constructed from Velodyne-64 LiDAR scans. Table 1 shows the average recall at top 1% and at top 1 for each method on Oxford and in-house benchmark. Our method performs better than others for networks with NetVLAD global feature extraction regardless of selecting several or only one loop closure candidate. Our method achieves the best performance among all the data sets we did not train on. For methods that use GeM pooling, MinkLoc3D performs better on Oxford but performs similarly on U.S., R.A., and B.D. compared to the proposed E^{2}PN-GeM method.

Table 1: Experimental result showing the average recall (%) at top 1% and at top 1 for each of the methods on Oxford and in-house benchmark. Scan Context and M2DP are non-learning methods. Three methods in the middle rows use NetVLAD as a global pooling method. The last two methods in the bottom rows use GeM as a global pooling method.

	Oxford		U.S.		R.A.		B.D.	
	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1
Scan Context [19]	32.91	22.89	75.96	65.06	66.40	53.69	50.90	44.57
M2DP [18]	34.69	23.14	45.03	32.41	44.62	34.34	39.34	32.95
PointNetVLAD [15]	84.94	71.39	80.79	65.33	73.86	61.83	69.29	61.78
EPN-NetVLAD	89.17	77.69	87.82	74.03	81.98	70.09	76.91	69.14
E^2PN-NetVLAD	**93.78**	**85.04**	**92.85**	**83.19**	**87.23**	**79.36**	**86.82**	**81.83**
MinkLoc3D [23]	**97.91**	**93.76**	95.04	86.01	**91.19**	81.17	**88.48**	82.66
E^2PN-GeM	94.76	87.45	**95.36**	**88.47**	88.64	**82.39**	88.21	**83.29**

Table 2: KITTI experimental result shows the average recall (%) at top 1% for each model. All methods are only trained on Oxford. KITTI sequence 00 consists of loop closure in the same direction, whereas KITTI sequence 08 consists of loop closure in a reverse orientation.

	KITTI Sequence 00		KITTI Sequence 08	
	AR@1%	AR@1	AR@1%	AR@1
PointNetVLAD [15]	73.18	17.61	32.47	70.68
EPN-NetVLAD (Ours)	78.21	37.69	**63.84**	61.90
E^2PN-NetVLAD (Ours)	**79.45**	**43.40**	61.63	**71.43**
MinkLoc3D [23]	28.07	4.01	17.30	3.50
E^2PN-GeM (Ours)	**80.45**	**71.18**	**68.55**	**54.09**

4.2.2 KITTI benchmark

In addition to the above evaluation, we also evaluate the proposed methods on KITTI odometry data set [16]. 3D point clouds in the KITTI data set are collected by Velodyne HDL-64E, random downsampled to 4096 points, and scaled to [-1, 1] with zero mean. Different from data in Oxford, the points of ground are not removed. We choose sequence 00 and sequence 08 for evaluation. Sequence 00 has the highest number of scans and pairs for loop closure in the same orientation. Sequence 08 contains 100% reverse loop closure where there are revisiting the same place with 180-degree viewing angle differences and provides a more challenging scenario. For sequence 00, the first 170 seconds construct the reference database, and the remaining part of the sequence is used as test queries. Similarly, for sequence 08, the first 85 and middle 259 to 264 seconds construct the reference database, and the rest of the sequence is used as test queries. We ignore two nearby frames to avoid matching consecutive scans falsely. Table 2 reports the average recall at the top 1% and top 1 for place recognition in sequence 00 and sequence 08. All methods are trained using the same Oxford training data set. The table shows that the SE(3)-invariant property in EPN-NetVLAD, E^2PN-NetVLAD, and E^2PN-GeM helps them perform better in these challenging scenarios, supporting the better generalization claim.

4.2.3 Data Augmentation Experiment

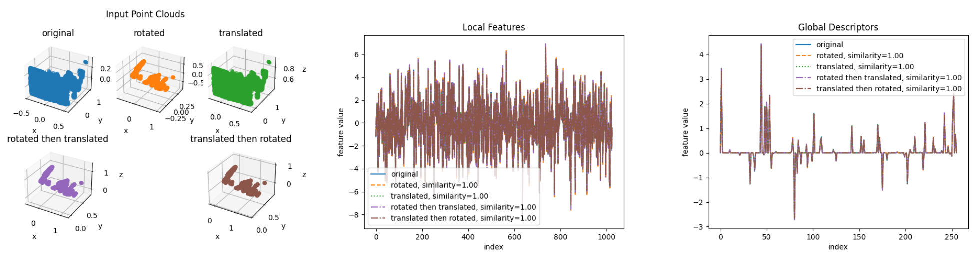
In Table 3, we experiment with different amount of training data. PointNetVLAD relies on both random transformation and increasing training data size to achieve high performance. Whereas E^2PN-NetVLAD can achieve similar performance with only training on three sequences. MinkLoc3D performs the best among all methods. However, it still requires random transformations in the training data.

4.3 Experiment with SE(3) Transformation

In addition to the place recognition experiments on Oxford and in-house benchmark, we construct simulated data to test the model performance with severe rotation and translation. First, we visualize

Table 3: Experimental result of data augmentation in training data size and if random transformation is applied during training.

	Random Transformation	Training Size: 3 Sequences		Training Size: 45 Sequences	
		AR@1%	AR@1	AR@1%	AR@1
PointNetVLAD [15]		69.38	54.00	86.88	73.12
PointNetVLAD [15]	✓	80.85	65.55	84.94	71.39
EPN-NetVLAD (Ours)		75.15	57.51	89.17	77.69
E^2PN-NetVLAD (Ours)		85.16	70.61	93.78	85.04
MinkLoc3D [23]	✓	-	-	97.91	93.76
E^2PN-GeM (Ours)		88.49	76.73	94.76	87.45



(a) point clouds under transformations

(b) SE(3)-invariant local features

(c) SE(3)-invariant global features

Figure 3: Visualization of the input point clouds, local features, and global descriptors under different transformations.

the local features and global descriptors when the input point cloud is transformed under rotation, translation, rotated then translated, and translated then rotated. Figure 3 shows the results and the cosine similarity score between each transformed feature/descriptor and the original feature/descriptor. We can see that even if the point cloud is rotated or translated, the output features and descriptors remain the same and has 100 % similarity to the original one.

Furthermore, we construct a simulated data set to include place recognition examples of different transformations. It contains the point clouds that are transformed under purely SO(3)-rotation, purely 3D translation, and with both rotation and translation. With original point clouds in a range between [-1, 1], 3D rotations are applied randomly, and 3D translations are applied with a standard deviation of 1.0. We use 440 point clouds, where each of them has two positive pairs. We then use the model trained on the Oxford benchmark to perform place recognition on this simulated data set. The result is shown in Table 4, EPN-NetVLAD performs significantly better in severe transformation. E^2PN's rotation-invariant property is not fully carried by NetVLAD and GeM. However, it still performs better than MinkLoc3D and PointNetVLAD.

4.4 Attentive Downsampling

We design an experiment to test the performance of the downsampling point cloud using an attention mechanism. Following the place recognition task experiment, we study the proposed network's performance with random and attentive downsampling methods. In this experiment, the network is constructed with only one layer of EPN with 64 local features and trained on three sequences of the Oxford data set to simplify the task. The result of different downsampling rates is presented in Table 5. It shows that using an attention mechanism to downsample point clouds can maintain high place recognition performance up to 50 % downsampling rate.

4.5 Run Time Performance

We tested our method on a system equipped with Intel i9-10900K CPU with a 3.7 GHz processor and an Nvidia GeForce RTX 3090. We also record the number of parameters in the network. For 3D point clouds with 4096 points, Table 6 shows the run time performance. E^2PN-GeM has the lowest number of parameters. PointNetVLAD and MinkLoc3D have the shortest inference time.

Table 4: Experimental result reports average recall at top 1% for performing place recognition task on different scenes where the point clouds are transformed under rotation or/and translation.

Rotation	Translation	PointNetVLAD	EPN-NetVLAD	E^2PN-NetVLAD	E^2PN-GeM	MinkLoc3D
✓		6.60 %	**98.74 %**	76.16 %	78.17 %	13.71 %
	✓	3.21 %	**100.00 %**	**100.00 %**	**100.00 %**	**100.00 %**
✓	✓	2.96 %	**99.43 %**	75.03 %	77.42 %	13.77 %

Table 5: Experimental result showing the average recall (%) at top 1% of EPN-NetVLAD when the input point cloud is downsampled with different percentages and different methods. This table compares the result of random downsampling and attentive downsampling, which utilize the attention mechanism to downsample.

Number of Points	Downsampling Rate	Random Downsampling	Attentive Downsampling
4096	0 %	71.66 %	71.66 %
3000	27 %	63.34 %	**71.65** %
2048	50 %	57.29 %	**71.05** %
1600	61 %	53.19 %	**66.22** %
1024	75 %	43.17 %	**57.97** %

Changing the global descriptor extraction method from NetVLAD to GeM drastically decrease the number of parameters but does not affect the run time substantially. We conjecture that the higher run times of SE(3)-equivariant networks are caused by the lack of network optimization. EPN and E^2PN are coded with custom functions to perform separate convolution, while other networks have network structures optimized on GPU. Thus, it is possible that SE(3)-equivariant networks can be further optimized in the future to improve run time.

5 Limitation

The major limitation of the proposed framework is the relatively slow run time and the need for optimized libraries to perform real-time place recognition. However, with the development of more powerful computing hardware, we expect this limitation to be largely resolved in the near future. In addition, the study of equivariant encoders under other Lie groups to enable invariance to, e.g., scale and deformation is an interesting future direction that we did not discuss in this paper.

6 Conclusion

We have designed a place recognition framework that exploits SE(3)-equivariant representation learning. In particular, SE(3)-invariant features learned from 3D point clouds improve robustness to large transformations and generalizability in place recognition tasks. In addition, we propose using an attention mechanism in place recognition to downsample the input point cloud while maintaining high performance. Our experimental results on real-world data sets show the proposed method performs well in various metrics. Future work includes a lightweight design of the equivariant encoder for real-time onboard applications and the extension of this framework to stereo cameras where image data can also be incorporated into the learned representation.

Table 6: Run time performance of the proposed framework and other learning-based place recognition methods. The input point cloud contains 4096 points. The run times are computed without any network optimization. *Prior to EPN-NetVLAD, attentive downsampling is performed to reduce point cloud size to 2048 points.

	Parameters	Run Time per Point Cloud (s)
PointNetVLAD [15]	19,779,145	0.006
MinkLoc3D [23]	1,055,713	0.005
EPN-NetVLAD (Ours)*	17,135,376	2.052
E^2PN-NetVLAD (Ours)	17,167,488	0.079
E^2PN-GeM (Ours)	192,513	0.082

References

[1] T. Barros, R. Pereira, L. Garrote, C. Premebida, and U. J. Nunes. Place recognition survey: An update on deep learning approaches. *arXiv preprint arXiv:2106.10458*, 2021.

[2] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. 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.

[3] M. Sanfourche, V. Vittori, and G. Le Besnerais. eVO: A realtime embedded stereo odometry for MAV applications. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2107–2114. IEEE, 2013.

[4] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *2013 IEEE International Conference on Robotics and Automation*, pages 54–61. IEEE, 2013.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[6] X. Zhang, L. Wang, and Y. Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.

[7] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu. RINet: Efficient 3D lidar-based place recognition using rotation invariant neural network. *IEEE Robotics and Automation Letters*, 7(2): 4321–4328, 2022.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[9] W. Zhang and C. 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.

[10] H. Chen, S. Liu, W. Chen, H. Li, and R. Hill. Equivariant point network for 3D point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021.

[11] M. Zhu, M. Ghaffari, W. A. Clark, and H. Peng. E^2PN: Efficient SE(3)-equivariant point network. *arXiv preprint arXiv:2206.05398*, 2022.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[13] F. Radenović, G. Tolias, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[14] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The Oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[15] M. A. Uy and G. H. Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4470–4479, 2018.

[16] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[17] T. Röhling, J. Mack, and D. 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.

[18] L. He, X. Wang, and H. Zhang. M2dp: A novel 3D point cloud descriptor and its application in loop closure detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 231–237. IEEE, 2016.

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

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

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

[22] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer. OverlapNet: Loop closing for LiDAR-based SLAM. In *Proceedings of the Robotics: Science and Systems Conference*, 2020.

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

[24] D. Cattaneo, M. Vaghi, and A. Valada. LCDNet: Deep loop closure detection for LiDAR SLAM based on unbalanced optimal transport. *arXiv preprint arXiv:2103.05056*, 2021.

[25] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong. 3D lidar-based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1380–1392, 2019.

[26] G. Kim, S. Choi, and A. Kim. Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 2021.

[27] Y. Fan, X. Du, and J. Shen. Fresco: Frequency-domain scan context for lidar-based place recognition with translation and rotation invariance. *arXiv preprint arXiv:2206.12628*, 2022.

[28] Y. Wang, Z. Sun, C.-Z. Xu, S. Sarma, J. Yang, and H. Kong. Lidar iris for loop-closure detection. *arXiv preprint arXiv:1912.03825*, 2019.

[29] D. Xu, J. Liu, J. Hyyppä, Y. Liang, and W. Tao. A heterogeneous 3D map-based place recognition solution using virtual lidar and a polar grid height coding image descriptor. *ISPRS Journal of Photogrammetry and Remote Sensing*, 183:1–18, 2022.

[30] C. Masone and B. Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021.

[31] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen. OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition. *IEEE Robotics and Automation Letters*, 2022.

[32] S. Lu, X. Xu, H. Yin, R. Xiong, and Y. Wang. One ring to rule them all: Radon sinogram for place recognition, orientation and translation estimation. *arXiv preprint arXiv:2204.07992*, 2022.

[33] P. Yin, F. Wang, A. Egorov, J. Hou, J. Zhang, and H. Choset. Seqspherevlad: Sequence matching enhanced orientation-invariant place recognition. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5024–5029. IEEE, 2020.

[34] Z. Fan, Z. Song, W. Zhang, H. Liu, J. He, and X. Du. Rpr-net: A point cloud-based rotation-aware large scale place recognition network. *arXiv preprint arXiv:2108.12790*, 2022.

[35] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning SO(3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision*, pages 52–68, 2018.

[36] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[37] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11348–11357, 2021.

[38] L. Wiesmann, R. Marcuzzi, C. Stachniss, and J. Behley. Retriever: Point cloud retrieval in compressed 3D maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2022.

[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[40] T. D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[41] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[42] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.