Multi-Modal Geometric Learning for Robot Localization

by

Chien Erh Lin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in the University of Michigan
2025

Doctoral Committee:

Assistant Professor Maani Ghaffari, Chair
Professor Jessy Grizzle
Assistant Professor Katherine Skinner
Professor Stella Yu

Chien Erh Lin

chienerh@umich.edu

ORCID iD: 0000-0001-6946-5920

© Chien Erh Lin 2025

To my loving husband, Tzu-Yuan (Justin)

Acknowledgments

I would like to begin by expressing my gratitude to my advisor, Prof. Maani Ghaffari, for his encouragement and guidance throughout my PhD journey. He supported me during challenging times, whether projects were stuck or papers were rejected, and allowed me to grow as a researcher.

I am also grateful to Prof. Jessy Grizzle for his support during my time at the University of Michigan, starting from my Master's program. My thanks go to Prof. Katie Skinner for her valuable feedback and advice and Prof. Stella Yu for giving me the computer vision perspectives.

I sincerely appreciate my collaborators. I am thankful to Dr. Ray Zhang, who mentored me in the early stages of my PhD, and to Dr. Minghan Zhu, who guided me later and shared his expertise in geometric and equivariant learning. I also want to thank my collaborators, Dr. Jingwei Song, Yanbaihui (Evelyn) Liu, Wan-Yi (Wendy) Yu, Avery Xi, and Samuel Ansaldo. I am incredibly grateful to the Computational Autonomy and Robotics Laboratory for their helpful discussions and support of my lab social ideas. To Dr. Tribhi Kathuria and Dr. Tzu-Yuan (Justin) Lin, thank you for the company of joining the program and sharing the journey with me.

I thank Michigan Robotics and Naval Architecture and Marine Engineering (NAME) for providing a supportive and collaborative environment. There are many people to thank, but I am incredibly grateful to Denise Edmund for helping with many things. I also appreciate Women+ in Robotics and Engineering (WiRE+) and thank Liz and Alia for starting WiRE+ with me and making great memories. I also thank the Taiwanese group in Mechanical Engineering and Robotics for their advice on classes and job opportunities.

I would also like to thank all the friends who supported me throughout my time in Michigan. Especially Liz, Grant, Tribhi, Alia, Shih-Chi, Kuan-Chieh, Mu-Ti, Yen-Tien (Den), Jia-Yi (Joy), Yi-Cheng, Chin-Wei (Pike), Yu-Wei (Katrina), Po-Kang (Brian), Hsuan-Cheng, Yu-Wei, and Chi-Kuan. Lastly, I would like to thank my family, Justin, and my cat, Hinoki, for your endless love and support.

TABLE OF CONTENTS

Dedication . . . . . **ii**

Acknowledgments . . . . . **iii**

List of Figures . . . . . **vii**

List of Tables . . . . . **ix**

Abstract . . . . . **xi**

Chapter

1 Introduction **1**

1.1 Contributions 2
1.2 Open-Sourced Software 3

2 Background & Related Works **4**

2.1 Geometric Learning 4
2.1.1 Equivariance and Invariance 4
2.2 Robot Localization 5
2.3 Robot Localization - Place Recognition 5
2.3.1 Exploiting Equivariance in Place Recognition 6
2.3.2 Attention Mechanism in Place Recognition 7
2.4 Robot Localization - Point Cloud Registration 7
2.4.1 Partial Overlap Point Cloud Registration 8
2.4.2 Exploiting Equivariance in Point Cloud Registration 9
2.5 Geometric Learning for 3D Point Clouds 9
2.5.1 Multi-Modal Data Fusion 10
2.5.2 Visual Foundation Models 11
2.5.3 Summary of Gaps in the Existing Literature 13

3 SE(3)-Equivariant Point Cloud-Based Place Recognition **14**

3.1 Introduction 14
3.2 Methodology 16
3.2.1 Attentive Downsampling 17
3.2.2 Local SE(3)-Equivariant Features 18
3.2.3 Local SE(3)-Invariant Feature Pooling 19
3.2.4 Global SE(3)-Invariant Place Representation 20

3.3 Experimental Results and Discussion . . . 21
3.3.1 Model Training . . . 21
3.3.2 Place Recognition Evaluation . . . 22
3.3.3 Place Recognition on Oxford Benchmark . . . 22
3.3.4 Place Recognition on In-House Benchmark . . . 23
3.3.5 KITTI benchmark . . . 23
3.3.6 Attentive Downsampling . . . 26
3.3.7 Data Augmentation Experiment . . . 27
3.3.8 Descriptors with SE(3)-Invariant Property . . . 27
3.3.9 Oxford Experiment with Random SE(3) Transformations . . . 29
3.3.10 Run Time Performance . . . 29
3.4 Limitation . . . 30
3.5 Conclusion . . . 31

4 SE3ET: SE(3)-Equivariant Transformer for Low-Overlap Point Cloud Registration . . . 32

4.1 Introduction . . . 32
4.2 Preliminary . . . 35
4.2.1 Equivariance and Invariance . . . 35
4.2.2 E2PN . . . 36
4.3 Methodology . . . 37
4.3.1 Equivariant Feature Encoder and Decoder . . . 37
4.3.2 Equivariant Transformer Design . . . 38
4.3.3 Equivariant Transformer Configurations . . . 43
4.3.4 Registration and Estimate Transformation . . . 44
4.4 Experimental Results . . . 44
4.4.1 Indoor Benchmark: 3DMatch/3DLoMatch . . . 45
4.4.2 Outdoor Benchmark: KITTI Point Clouds . . . 51
4.4.3 Generalization Experiment . . . 52
4.4.4 Run-time Analysis . . . 53
4.5 Conclusion . . . 53

5 Multi-Modal Fusion for Place Recognition: Integrating Visual Foundation Models with 3D Equivariant Features . . . 54

5.1 Introduction . . . 54
5.2 Methodology . . . 57
5.2.1 Visual Foundation Model Image Features . . . 57
5.2.2 Equivariant Point Cloud Features . . . 58
5.2.3 Fusing Approaches . . . 59
5.3 Experiments . . . 62
5.3.1 HPointLoc Data Set . . . 62
5.3.2 Oxford RobotCar Data Set . . . 64
5.3.3 KITTI Odeomtry Data Set . . . 64
5.3.4 Invariance Experiment . . . 65
5.4 Limitation . . . 66

5.5 Conclusion 68
6 Conclusion and Future Works 69
Bibliography 71

List of Figures

FIGURE

3.1 Overview of the proposed SE(3)-equivariant point cloud-based place recognition pipeline. Optionally, 3D point clouds are preprocessed with attentive downsampling. Next, SE(3)-equivariant local features are learned using SE(3)-equivariant networks. In this work, we leverage EPN and E2PN. 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. 17

3.2 Place recognition evaluation on Oxford benchmark of the proposed methods and state-of-the-art approaches MinkLoc3D [42], PointNetVLAD [69], M2DP [33], and Scan Context [40]. 24

3.3 Visualization of one input point cloud, the same point cloud under transformations, their corresponding local features, and the output global descriptors. There are 1024 local features for each point in the point cloud. We only show the first 100 features of the first point for this visualization. 28

4.1 SE3ET can register two low-overlap point clouds with significant rotations and translations. This qualitative result is performed on rotated 3DLoMatch, where the first row is an easy example (28.72 % overlapping ratio with multiple overlapping surfaces), the middle row is a moderate example (10.51 % overlapping ratio with multiple overlapping surfaces), and the last row is a challenging example (26.75 % overlapping ratio with only one overlapping surface). 34

4.2 The proposed point cloud registration framework includes a SE(3)-equivariant feature encoder and decoder and an equivariant transformer design for learning the point correspondences of superpoints. The dark blue blocks and arrows are equivariant, and the green blocks and arrows are invariant. We propose two transformers structures *SE3ET-E* and *SE3ET-I*. Here, SA stands for self-attention, and CA stands for cross-attention. 35

4.3 The geometric (octahedron shape) and algebraic (color bricks) illustration of using permutation to recover discretized rotation group. Each color represents the feature of one anchor (vertex), and the different order of the combination of features represents the discretized rotation defined in the network. If the octahedron is rotated 90 degrees clockwise, the order of the features in the anchor dimension changes accordingly. The discretization of the rotation groups is derived from the permutation of the discrete anchors. For $A = 6$, the rotation group contains 24 rotations. 36
4.4 An illustration of the equivariant anchor-based cross-attention (ACA) and rotation-based cross-attention modules (RCA). 40

5.1 Differences between two modalities: image and point cloud. These images and point clouds are from the Oxford RobotCar [54] (top), KITTI odometry [29] (middle), and HPointLoc [91] (bottom), where each pair of input are from the same place. Point cloud data remain consistent despite illumination and viewpoint changes, whereas image data provide additional information, such as road lane lines and paintings in the room. 56
5.2 Different fusion methods. Direct sum combines two modalities when they are both global representations. Invariant fusion integrates two modalities on a local scale. Equivariant fusion preserves the equivariant features from 3D data and obtains global representation at a later stage. 60
5.3 Place recognition performance on HPointLoc dataset. 64
5.4 Average recall curve for performing place recognition on Oxford RobotCar dataset sequence 5, 6, and 7 using the direct sum approach. The multi-modal result is better than any single-modal. 65
5.5 Average recall curve for performing place recognition on KITTI sequence 08 using the direct sum approach. The multi-modal result is better than any single-modal method. 66
5.6 Images and point clouds from the HPointLoc data set after rotation for 90, 180, and 270 degrees. 67
5.7 Results of the invariance experiment. Equivariant fusion is performing better than other methods. 67

List of Tables

TABLE

3.1 Experimental result showing the average recall (%) at top 1% and at top 1 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. * is with attentive downsampling. 25

3.2 Average recall (%) at top 1% on KITTI sequence 00 and 08. * is with attentive downsampling. 26

3.3 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 on Oxford benchmark. 26

3.4 Data augmentation experiment on Oxford benchmark with different training data size and whether random transformation is applied during training. * is with attentive downsampling. 27

3.5 Average recall (%) at top 1% for performing place recognition task on simulated data where the point clouds are transformed under rotation or/and translation. * is with attentive downsampling. 29

3.6 Run time performance per point cloud which contains 4096 points. The run times are computed without any network optimization. * attentive downsampling is performed to reduce point cloud size to 2048 points. 30

4.1 Evaluation on the original and rotated 3DMatch/3DLoMatch benchmark. Excluding the methods using extra information, the best result is shown in **bold**, and the second best is shown in the underline. SE3ET-E2 and SE3ET-I2 are the same as our proposed structures, with only half of the feature size. 48

4.2 Registration recall and inlier ratio evaluation results on 3DMatch and 3DLoMatch for different number of samples. RANSAC is run with 1k iterations for YOHO and RoReg and 50k iterations for other methods. * symbol means the method uses extra information, which is excluded from the ranking. The best result is in **bold**, and the second is in the underline. 49

4.3 Evaluation results on rotated 3DMatch and 3DLoMatch for different number of samples. RANSAC is run with 1k iterations for YOHO, as stated in their paper, and 50k iterations for other methods. The best result is shown in **bold**. 50

4.4 Registration results on 3DMatch in absolute metrics Relative Translation Error (cm) and Relative Rotation Error (°). The best result is shown in **bold**. 50

4.5 Evaluation result on KITTI point clouds. The best result is shown in **bold**. . . 51
4.6 Result for the generalization experiment. Evaluation results on KITTI point clouds with models trained on 3DMatch. The best result is shown in **bold**. . . . 51
4.7 Runtime analysis of the proposed method. We tested on an Intel i9-10900K CPU and Nvidia GeForce RTX 3090 GPU. 52

5.1 Place recognition results on three data sets. 63

Abstract

Robot localization uses external sensor data to estimate a robot's position, but movements can alter vision measurements and learned features, impacting robustness. This thesis introduces a new perspective on multi-modal robot localization by integrating geometric learning, primarily focusing on equivariant networks to address these challenges.

First, we explore the application of geometric learning in the loop closure process, enabling the robust identification of previously visited locations despite transformative changes. This framework's SE(3)-invariant attribute offers stable, consistent descriptors regardless of changes in robot pose.

Next, we propose SE(3)-equivariant transformer designs for learning point cloud correspondences, crucial for point cloud registration tasks. Through adopting equivariant feature learning, the framework significantly improves the robustness of point cloud registration with low overlap and significant pose changes.

Finally, we integrate geometric learning with visual foundation models for localization tasks. The proposed fusing methods bridge the gap that combining images with equivariant point cloud features while preserving their equivariance is challenging due to the group structure that is added to the network architecture.

Overall, this dissertation explores the benefits of incorporating geometric constraints into machine learning models, significantly enhancing both the reliability and performance of robot localization and contributing to critical advancements in robotics.

Chapter 1

Introduction

Robot localization is essential in robot autonomy in various applications, from autonomous vehicles and mobile service robots to search and rescue robots or Unmanned Aerial Vehicles. This process determines a robot's position and orientation within its environment, allowing it to know its location in the world coordinate. Such localization is vital for downstream tasks such as map building, motion planning, and environmental navigation. Notable advancements have been made in this domain, especially in adapting images and 3D point clouds for robot localization. The evolution of advanced sensors and computational hardware has enabled robots to use images and point clouds onboard, contributing rich data for higher accuracy and more robust models.

Machine learning, specifically deep learning algorithms, has demonstrated the potential to enhance robot localization. These algorithms empower robots to predict their locations based on features they learn from images or point clouds. This tool is especially beneficial when deriving the robot's pose from sensory data is difficult due to noise disturbances, occlusion, or featureless scenes. Thanks to these capabilities, learning-based methods offer substantial potential in handling challenging scenarios and facilitating more practical uses for robots in real-world applications.

However, the system's stability faces potential compromise when robot pose transformations bring changes in image and point cloud measurements. Incorporating geometric information as an inductive bias into the learning algorithm can help the

localization algorithm maintain its performance even with these disturbances. One way to achieve this is through leveraging the properties of invariance and equivariance in machine learning. Invariance represents a model where the output is independent of the transformation that applies to the input, while equivariance preserves the transformations from the input to output data. Such properties, therefore, improve our capacity to manage sensor data transformations more effectively.

In this dissertation, we propose to bring value to robot localization algorithms by integrating geometric constraints into machine learning models. We examine this concept across various localization tasks involving multiple modalities, including point cloud registration, place recognition, and orientation estimation. We elaborate on the benefits and applications of geometric learning in robotics. It not only aids in overcoming the limitations of traditional methods but also paves the way for advancements within robotics, offering immense value in real-world environments.

1.1 Contributions

The main contributions of this thesis are summarized as follows:

- In Chapter 2, a background of related works is introduced. We introduce existing robot localization works focusing on place recognition and point cloud registration, geometric learning on 3D data, visual foundation models, and multi-modal fusion methods.

- In Chapter 3, we proposes a framework for SE(3)-equivariant point cloud place recognition. We introduce using equivariant networks for robust point cloud place recognition against arbitrary rotation and translation of robot poses. We use equivariant features to learn invariant point cloud representations. The proposed methods are generalizable and scalable to unseen data sets, removing the need for data augmentation.

- In Chapter 4, we proposes a low overlap point cloud registration framework using SE(3)-equivariant feature learning and equivariant transformer. We propose various designs of equivariant transformers, each offering unique properties and potential uses for finding point correspondences. Our feature learning based on equivariant convolutions and transformers improves the robustness against point clouds with low overlap and significant pose changes for point cloud registration.

- In Chapter 5, a multi-modal fusion framework is proposed. We introduce various approaches to integrate 3D features from geometric learning and 2D features from visual foundation models to provide robust and stable representation.

1.2 Open-Sourced Software

Open-source implementation of the proposed algorithms is provided as follows:

- `https://github.com/UMich-CURLY/se3_equivariant_place_recognition`

- `https://github.com/UMich-CURLY/SE3ET`

- `https://github.com/UMich-CURLY/equivariant_foundation_model_fusion`

Chapter 2

Background & Related Works

We first introduce geometric learning and then review robot localization tasks such as place recognition and point cloud registration. Next, we discuss geometric learning networks for 3D point clouds and their effectiveness in addressing generalization challenges. To study how to integrate geometric learning into a multi-modal framework, we introduce the concept of multi-modal data fusion and its significance in current research. Following this, we investigate visual foundation models, examining their strengths and highlighting their limitations in using 3D data. Finally, we summarize the research gap in the existing works.

2.1 Geometric Learning

Geometric learning is a domain within deep learning that focuses on embedding geometric structures into the learned features. In this thesis, we specifically address transformations such as rotation and translation as the core geometric information to be embedded by the network. Below, we introduce two key concepts that highlight the capabilities of geometric learning.

2.1.1 Equivariance and Invariance

Invariance is where the output of a function remains unchanged under transformations of the input, expressed as $f_{inv}(g \circ x) = f_{inv}(x)$. This is useful for tasks like classification, where different input transformations represent the same object.

Equivariance is a more generalized form where the function's output undergoes the same transformation as the input, expressed as $f_{equ}(g \circ x) = g \circ f_{equ}(x)$. Equivariant representations are more expressive than invariant representations, offering better cross-domain generalizability.

2.2 Robot Localization

Robot localization estimates the robot's position and orientation, given sensor measurements and the world model. When the sensor measurements are 3D point clouds, point cloud registration is often used to estimate the transformation between two measurements [61]. By estimating the transformations, the accumulated trajectory of the robot can be obtained. If the world model, or a map, is known for the robotic system, loop closure can be beneficial to eliminate the accumulated error from the trajectory. Place recognition, also known as loop closure detection, determines if the current measurement has been seen before and is critical to the Simultaneous Localization and Mapping (SLAM) system. Place recognition reduces the error of the trajectory and map, resulting in a more consistent and accurate map [77]. In the following sections, we introduce the existing works in place recognition and point cloud registration.

2.3 Robot Localization - Place Recognition

Previously, place recognition using 3D point clouds relied on histograms or hand-engineered features such as Fast Histogram [66], M2DP [33] and Scan Context [40]. M2DP [33] projects 3D point clouds to multiple 2D planes and constructs global descriptors from singular vectors of density signatures. Scan Context [40] represents the point cloud in the polar axis and encodes the height of the observed points into the representation. PointNetVLAD [69] is a pioneering work to apply a learning-based feature extractor to place recognition tasks. It combines local feature

encoder PointNet [62] and global descriptor aggregator NetVLAD [3] to allow end-to-end representation training from a given 3D point cloud. Later, MinkLoc3D [42] presents sparse voxelized point cloud representation and sparse 3D convolutions. Other algorithms such as LPD-Net [50], OverlapNet [15], and LCDNet [9] provide different learning-based ways of constructing global descriptors for place recognition tasks. However, these algorithms are not robust to rotational and translational pose changes.

2.3.1 Exploiting Equivariance in Place Recognition

Considering that the observer may be in different orientations or locations, researchers propose some rotation-invariant hand-crafted descriptors such as histograms of range in a LiDAR scan [86], Scan Context [40, 39], frequency-domain Scan Context [24], LiDAR-Iris [74], and height coding descriptor [82] to perform place recognition.

Yin, Huan, et al. [86] 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 [40] 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++ [39] to include lateral invariance by augmentation based on urban road assumption. FreSCo [24] uses frequency-domain Scan Context to perform place recognition with translation and rotation invariance. LiDAR-Iris [74] encodes height information into eight-bit binary code and uses the Fourier transform to estimate the translation between two LiDAR-Iris images to remove the rotational difference between LiDAR scans. Xu, Dong, et al. [82] 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 [55].

Only a few works try to encode rotation-invariant or translation-invariant features into learning-based place recognition algorithms. PointNetVLAD [69] and LCDNet [9] try to increase robustness by randomly rotating input point clouds during training. To achieve yaw-angle-invariant, OverlapNet [15] and OverlapTransformer [53] use range images and [52] propose a RING descriptor. To achieve rotation-invariant, RINet [44] exploits additional semantic information with rotation equivariant convolution, Seq-SphereVLAD [87] uses spherical convolution in spherical view, and RPR-Net [25] exploits rotation-invariant convolution. Nevertheless, these strategies do not consider both 3D rotation and translation differences in the pose; thus, they might not be sufficient in more challenging scenarios.

2.3.2 Attention Mechanism in Place Recognition

Attention [70] has been applied to some place recognition tasks to learn the contextual features. PCAN [96], SOE-Net [81], Retriever [78], and OverlapTransformer [53] include this technique to learn the relations of different features. Among these applications, attention mechanisms are used to learn the importance of local features. In chapter 3, we explore applying the attention mechanism to the input 3D points to learn the points' significance for downsampling.

2.4 Robot Localization - Point Cloud Registration

Traditional point cloud registration methods extract handcrafted local features to establish point-to-point correspondences. One of the pioneering works is the Iterative Closest Point (ICP) algorithm [6]. However, these methods often converge to local minima and are ineffective for large-scale data sets due to their reliance on geometric properties.

Learning-based methods have gained attention for their impressive performance. Correspondence-based methods [88, 63, 27, 75, 8, 84] follow a two-step process:

correspondence matching and finding the optimal transformation. First, they compute correspondences between source and target point clouds based on point distances or feature similarities. Then, they calculate the optimal rigid transformation (rotation and translation) to align the point pairs, using RANSAC [26], weighted SVD [6, 75], or LGR [63]. Although widely used, these methods are prone to incorrect correspondences. We propose using equivariant features to improve accuracy in our correspondence-based framework.

2.4.1 Partial Overlap Point Cloud Registration

A common challenge in the point cloud registration task is the partial overlap situation, where the overlapping ratio between the two point clouds is low, especially when dealing with noisy data, occlusions, or outliers. Various strategies have been developed, such as using a graph neural network to learn structural information [34, 76], learning posterior probability distributions of Gaussian mixture models (GMMs) from point clouds [56], or a transformer mechanism [70] to either implicitly or explicitly learn the overlapping points [76, 63, 100, 85]. Another approach is using features learned from coarse and fine scales of the point cloud to find overlapping patches and point correspondence [88, 63, 37]. On the other hand, systems like PEAL [90] use prior overlapping estimation in the registration process, which offers a unique approach to the issue but requires estimated transformation before performing registration. While these works progress in tackling the partial overlap challenge, when it comes to substantial rotational-wise transformation, most of the above methods rely on data augmentation during training to increase robustness to arbitrary transformation.

2.4.2 Exploiting Equivariance in Point Cloud Registration

Some studies, such as G3DOA [98], PPFNet [20], and SpinNet [2], craft rotation-invariant features. RoITr [89] and BUFFER [1] further extend SpinNet [2] module to perform rotation-invariant point matching. However, these crafted rotation-invariant features could be oversimplified, which may sacrifice expressiveness. On the other hand, equivariant representations preserve more expressive features, providing better cross-domain generalizability than invariant representations.

2.5 Geometric Learning for 3D Point Clouds

3D point clouds offer geometry information essential for real-world robot deployment. Despite their significance, research on 3D representation remains relatively understudied, compounded by the scarcity of large-scale 3D datasets necessary for the training of foundational models.

While the development of 3D foundation models is still ongoing, a promising approach in the field of 3D representation learning is equivariant networks, which have shown significant improvements in the generalizability of models.

Equivariant networks incorporate the symmetry structure of mathematical groups into the network for equivariant feature learning. Equivariant features preserve the transformations applied to the input in the output, while invariant features derived from equivariant features remain unchanged under these transformations.

A series of works design network architectures with the equivariance property for general feature learning. For example, Spherical CNNs [23] and Vector Neuron [19] enable SO(3)-equivariant representation learning. Equivariant Point Network (EPN) [13] performs SE(3) separable convolution, which separates 6D convolution into convolutions in the 3D Euclidean space and in SO(3). E2PN [101] proposes a lightweight variant of SE(3)-equivariant network for point clouds. Moreover, [28]

and [12] research into integrating equivariant learning within the transformer mechanism demonstrates its applicability to tasks such as classification and reconstruction.

These networks, which learn equivariant features from 3D point clouds, offer valuable insights into the network architecture for point cloud registration across various transformation scenarios. They also contribute to delivering a stable representation of a place under different transformations. Generally, these networks address the group-equivariant feature learning problem in classification and segmentation tasks. However, while they have generally been tested with point clouds of single object shapes, they have not been extensively evaluated on 3D point clouds in real-world outdoor scenes.

Since the transformation of the point cloud is preserved in the features learned by equivariant networks, these models demonstrate strong generalization performance and efficient training without the need for extensive data augmentation. This is particularly beneficial for tasks such as classification, segmentation, and place recognition [46].

YOHO [72], a notable example, employs rotation-equivariant feature learning for point cloud registration and has been extended to RoReg [73]. While these methods effectively leverage equivariant feature learning for point cloud registration, further optimization is needed, particularly in scenarios with low overlap.

2.5.1 Multi-Modal Data Fusion

Humans perceive the environment through various senses, and similarly, a single measurement instrument may not provide a complete understanding for machines to understand their surroundings [43]. Multi-modal data fusion leverages multiple types of sensors with complementary characteristics to enhance performance [18]. For instance, while images might struggle to distinguish objects with similar colors but different orientations, 3D point clouds can measure the structure of the environment,

overcoming this limitation.

Common multi-modal fusion methods can be categorized into three main types: early fusion, late fusion, and deep fusion.

Early Fusion: This approach fuses data at the raw data level, allowing for potential cross-modal interactions. However, challenges include aligning heterogeneous modalities, such as images in the projective space with point clouds in Euclidean coordinates. Additionally, disparities in sparsity between modalities can result in loss of information for the denser modality [80].

Late Fusion: This method integrates decisions from different modalities at the decision-making level. Late fusion does not require synchronization between modalities; it only needs to align results at the end. However, the reliance on limited information can affect the comprehensiveness of the outcome [58].

Deep Fusion: This approach involves fusing data at a higher level of abstraction, typically in the latent space, where machine-learned features are processed. Data in deep fusion is handled in a parallel manner, facilitating robust integration of multiple modalities [45, 7]. This is the approach we adopt because it ensures efficient data processing and higher accuracy by combining features at a higher abstraction level, where the most relevant information has already been extracted.

2.5.2 Visual Foundation Models

Foundation models have shown impressive zero-shot performance on various tasks. Here, we focus on visual foundation models that learn visual representations from images.

Most visual foundation models employ Vision Transformer (ViT) [22] as the image encoder. The ViT trains a transformer encoder on ImageNet and is capable of capturing global context in images. A notable example is Contrastive Language-Image Pretraining (CLIP) [65], a visual language model trained on 400 million

image-text pairs using a contrastive loss. MERU [21] advances CLIP with hyperbolic representations of images and text, allowing it to better capture underlying hierarchies in image-text datasets. ImageBind [30] introduces a shared representation space for cross-modal joint embedding that includes audio, depth, thermal, and IMU modalities, in addition to the image and text pairs for image retrieval. DINOv2 [57] distills large models into smaller ones, enabling more efficient implementations in downstream tasks.

Currently, no foundation model directly takes in point clouds due to the shortage of 3D pertaining data. The largest 3D single-object dataset, ShapeNet [11], comprises only about 51k 3D models, which falls short of the scale required for training a robust visual foundation model. In comparison, CLIP[65] was trained on 400 million images. Some approaches project 3D data onto 2D to leverage pre-trained 2D visual foundation models more effectively. For instance, SemAbs [31] uses RGB-D images and a relevancy map to predict 3D occupancy for objects. PointCLIP[95] converts point clouds into multi-view depth maps and compares them with text features from text input. Similarly, CLIP2point [35] renders images from 3D models and generates corresponding depth maps to train the depth encoder using an image encoder. However, projecting point clouds to depth maps may result in losing some geometric information.

Ongoing research continues to address the challenges in this area. PointLLM [83] uses a multi-layer perceptron (MLP) to project pre-trained point cloud encoder features to match the token dimension of a pre-trained large language mode backbone. CLIP^2 [93] introduces a proxy alignment mechanism to find correspondence between images and point clouds for classification. Nevertheless, these efforts are predominantly focused on object-level point clouds, which are inadequate for comprehensive real-world understanding. As research evolves, addressing the limitations of current methods and extending the capabilities of foundation models to effectively incorporate 3D

data remains a crucial area of study.

2.5.3 Summary of Gaps in the Existing Literature

The field of 3D geometric representation remains under-explored, and this thesis aims to investigate how geometric learning can be incorporated into robot localization tasks. In particular, for the place recognition task, we will propose a framework that is the first to develop an SE(3)-equivariant place recognition framework, thereby bridging the gap between group-equivariant literature and place recognition. Additionally, we will explore the design of transformer mechanisms that can handle the equivariant features learned from geometric processing for point cloud registration tasks. Furthermore, the fusion between visual foundation models and 3D equivariant features represents a promising direction for enhancing the understanding and representation of real-world environments. Despite their individual successes, there is a gap in leveraging multi-modal fusion to integrate visual foundation models with 3D equivariant features effectively. This thesis aims to bridge this gap by providing a framework for fusing equivariant point features with image features. Addressing this gap offers a pathway to creating more comprehensive, robust, and efficient systems capable of understanding the world through both visual and 3D data.

Chapter 3

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

In this chapter, we introduce 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 that 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. We have open-sourced our code at `https://github.com/UMich-CURLY/se3_equivariant_place_recognition`.

3.1 Introduction

Place recognition, also known as loop closure detection, can be defined as linking the sensor's in-situ observations and the prebuilt reference map. It is a critical component in Simultaneous Localization and Mapping (SLAM) and enables a robot to determine if it has seen a place before and provides loop closure candidates [38].

Among numerous robot perception sensors, 3D (stereo, LiDAR, and RGB-D) sensors are gaining popularity. They are widely equipped in modern service robots, autonomous cars [32], and drones [67] due to their better environment perception ability and decreasing prices. Thus, place recognition techniques with 3D data can be used to estimate the agent's location in scenarios such as self-driving vehicles, autonomous indoor navigation, or scientific exploration.

Extracting consistent features from 3D data is an important research topic but remains underexplored and unsolved [97]. 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 [44]. Thus, the extracted global descriptors and their place recognition performance are sensitive to transformation variations in the training and testing point cloud samples. Our research aims to design 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 [70] enables networks to learn the correlation between input features and obtain the importance of each feature. Some place recognition frameworks like PCAN [96] 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)-equivariant representation learning to perform place recognition in challenging rotation and translation scenes (Figure 3.1). We learn SE(3)-equivariant local features via group-equivariant encoder networks. And our pipeline is agnostic to the particular approach for learning equivariant features. Then, SE(3)-invariant global descriptors are learned by aggregating local features. Moreover, we apply a self-attention mechanism

for downsampling point clouds to decrease memory usage and increase efficiency in training. We train and test our network on real-world data sets. 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 at `https://github.com/UMich-CURLY/se3_equivariant_place_recognition`.

3.2 Methodology

This section details our proposed framework for SE(3)-invariant place recognition using 3D point clouds. Figure 3.1 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.

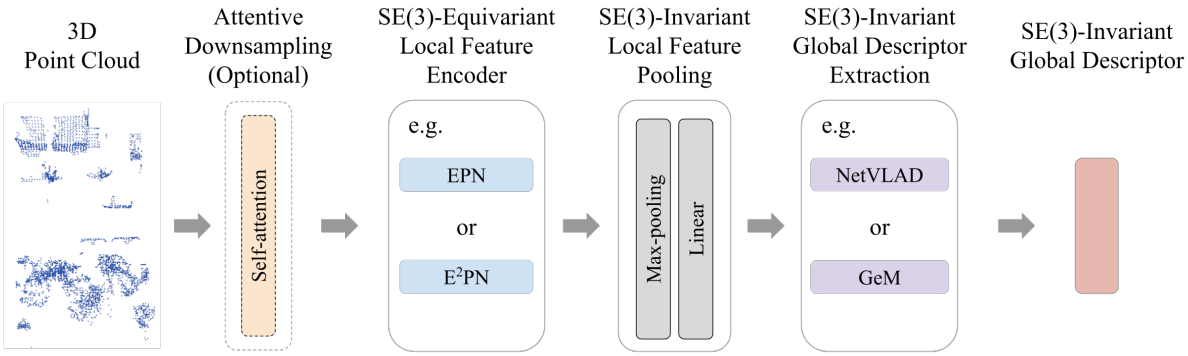


Figure 3.1: Overview of the proposed SE(3)-equivariant point cloud-based place recognition pipeline. Optionally, 3D point clouds are preprocessed with attentive downsampling. Next, SE(3)-equivariant local features are learned using SE(3)-equivariant networks. In this work, we leverage EPN and E2PN. Then, 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.

3.2.1 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 [70] using the PyTorch library [59] to learn the attention weights $W_{atten} \in \mathbb{R}^{N\times 3}$ from the input point cloud. Multi-head attention is defined as $\text{MultiHead}(X) = \text{Concat}(head_1, head_2, head_3)W^O$, where $\text{Concat}(\cdot)$ does the features concatenation. The number of parallel attention heads is set as 3. Each attention head is defined as $head_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V)$. 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 the 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 $P' = \mathbb{R}^{k\times 3}$.

3.2.2 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 $f_{inv} : X \to Y$ 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 $\mathbb{R}^3$ [5].

In this work, we leverage Equivariant Point Network (EPN) [13] and E2PN [101] to learn the SE(3)-equivariant feature and capture the inherent symmetry of 3D point cloud data. In the original EPN [13], given a 3D point x, a rotation g, a feature representation function $\mathcal{F} : \mathbb{R}^3 \times \mathrm{SO}(3) \to \mathbb{R}^D$, and a kernel $h : \mathbb{R}^3 \times \mathrm{SO}(3) \to \mathbb{R}^D$, the discretized SE(3)-equivariant convolutional operator is defined as the dot product between the translated and rotated kernel and the function $\mathcal{F}$: $(\mathcal{F} * h)(x, g) = \sum_{x_i \in \mathcal{P}} \sum_{g_j \in G} \mathcal{F}(x_i, g_j) h(g^{-1}(x - x_i), g_j^{-1} g)$, where P and G are the discretized sets

corresponding to $\mathbb{R}^3$ and 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 E2PN [101], a lightweight and more efficient variant of EPN [13]. E2PN 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 outdoor environments.

3.2.3 Local SE(3)-Invariant Feature Pooling

After learning from an SE(3)-equivariant encoder, we obtain SE(3)-equivariant features $f_e(P) \in \mathbb{R}^{N \times C \times R}$, where P is the input point cloud, $f_e(\cdot)$ is the mapping from point cloud to SE(3)-equivariant features, N is the number of points, C is the number of local features, and R is the number of rotation group discretization. We then apply the pooling method to extract SE(3)-invariant features. To avoid the group attentive pooling failing if the point cloud is circularly symmetric as discussed in [13], 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. In the max-pooling step, we only keep the maximum feature from one of the R discretized rotation group elements. 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 3.1 for an illustration.

3.2.4 Global SE(3)-Invariant Place Representation

Global descriptors are computed by aggregating local features using NetVLAD or Generalized Mean (GeM) [64]. NetVLAD learns cluster centers of VLAD (Vector of Locally Aggregated Descriptors) in a CNN framework. The output descriptors V are defined as Equation (3.1).

$$V(j,k) = \sum_{i=1}^{N} \frac{e^{w_k^\intercal x_i + b_k}}{\sum_{k'} e^{w_{k'}^\intercal x_i + b_{k'}}} (x_i(j) - c_k(j)). \tag{3.1}$$

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.

GeM is a trainable pooling layer that generalizes max-pooling and mean-pooling. With local feature input x, the GeM pooling output is defined as Equation (3.2)

$$f_{GeM} = (\frac{1}{|x|} \sum_{x_i \in x} x_i^p)^{\frac{1}{p}} \tag{3.2}$$

, where p is a pooling parameter that can be learned or set manually.

To learn discriminative global descriptors, we use lazy quadruplet loss [69] to train the network to perform place recognition tasks. For each iteration of training, there is an anchor point cloud P_a, a "positive" point cloud P_p that is similar to P_a, and a "negative" point cloud $\{P_n\}$ that are dissimilar to P_a, and a random point cloud in the training set P_{n^*}. The lazy quadruplet loss is defined as Equation (3.3)

$$Loss(P_a, P_p, P_n, P_{n^*}) = \max_j([\alpha + \delta_p - \delta_{n_j}]_+) + \max_k([\beta + \delta_p - \delta_{n_k^*}]_+) \tag{3.3}$$

, where α and β are constant values to provide margin. The lazy quadruplet loss can minimize the L2 distance between the anchor and the positive representation $\delta_p = d(f(P_a), f(P_p))$ while maximizing the distance between the anchor and the negative representation $\delta_{n_j} = d(f(P_a), f(P_{n_j})), P_{n_j} \in \{P_n\}$.

3.3 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, the SE(3)-invariant properties, and the design of attentive downsampling.

3.3.1 Model Training

We train our networks on Oxford RobotCar [54] benchmark created by [69]. 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 prebuilt map. The ground points are removed, and the point clouds are normalized to be zero mean and rescaled to within 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 sample, while a location more than 50 meters away is considered a negative sample. We train with 21,711 sub-maps. In EPN-NetVLAD, point clouds are first downsampled to 2048 points using the attention mechanism. For SE(3)-equivariant encoders (EPN and E2PN), we construct them with two layers of equivariant layers, one with 32 local features and one with 64 local features. The number of dimensions for the discrete rotation group is 60 in EPN and 12 in E2PN. The encoder is followed with max-pooling and a linear layer to map local features to 1024 dimensions. Then, we use NetVLAD or GeM to learn global descriptors with dimensions of 256. For GeM, we follow MinkLoc3D's structure and set the pooling parameter $p = 3$. The network is trained for 30 epochs with a learning rate of 5×10^{-5}. The hyper-parameters in lazy quadruplet loss are set as $\alpha = 0.5, \beta = 0.2$. The network parameters are optimized by ADAM [41].

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.

3.3.2 Place Recognition Evaluation

In place recognition tasks, precision and recall are the two well-established evaluation metrics [51]. Precision $= TP/(TP + FP)$ is the percentage of true loop closures among all the places we recognize. Recall $= TP/(TP + FN)$ is the percentage of places we recognize among all true loop closures. 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 defined as $\mathrm{F1} = 2(\mathrm{precision} \cdot \mathrm{recall})(\mathrm{precision} + \mathrm{recall})$ to obtain a balancing metric between precision and recall.

3.3.3 Place Recognition on Oxford Benchmark

We first evaluate the performance of the proposed method on the Oxford benchmark. The Oxford RobotCar data set 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 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 and recall rate, then 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 [60], 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 3.2. All learning-based methods are trained on the same Oxford

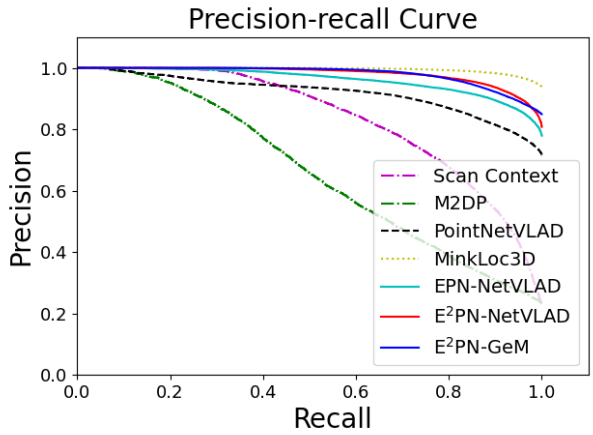
benchmark training set. However, MinkLoc3D is trained with a more efficient training strategy. Scan Context [40] and M2DP [33] construct hand-engineered features to perform place recognition. The result from Scan Context and M2DP can be further improved with hyper-parameter tuning. The figure shows that the proposed networks E2PN-NetVLAD and EPN-NetVLAD outperform PointNetVLAD, all of which share the same global feature extraction method. MinkLoc3D and E2PN-GeM both use GeM pooling for global feature extraction. Though MinkLoc3D performs the best among all methods, E2PN-GeM and E2PN-NetVLAD still perform consistently within 5% of difference.

3.3.4 Place Recognition on In-House Benchmark

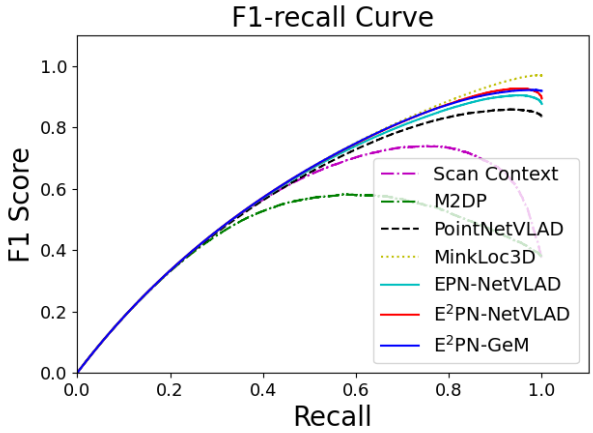
To show the generalizability of the proposed method, we also evaluate all methods on in-house data sets [69], which are constructed from Velodyne-64 LiDAR scans in three kinds of regions that are unseen to the network, including the university sector (U.S.), residential area (R.A.), and business district (B.D.). Table 3.1 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 candidates. 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 similarly on U.S., R.A., and B.D. compared to the proposed E2PN-GeM method, which shows the generalizability among data sets of the proposed method.

3.3.5 KITTI benchmark

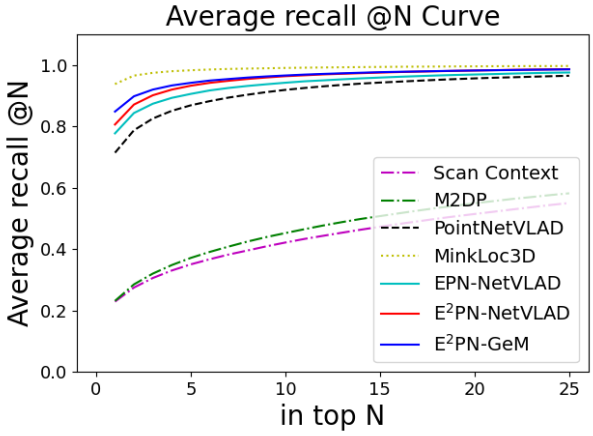
We also evaluate the proposed methods on KITTI odometry data set [29]. Different from data in Oxford, 3D point clouds in the KITTI data set are collected by



(a)



(b)



(c)

Figure 3.2: Place recognition evaluation on Oxford benchmark of the proposed methods and state-of-the-art approaches MinkLoc3D [42], PointNetVLAD [69], M2DP [33], and Scan Context [40].

Table 3.1: Experimental result showing the average recall (%) at top 1% and at top 1 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. * is with attentive downsampling.

		Oxford		U.S.		R.A.		B.D.	
		AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1	AR@1%	AR@1
hand-crafted	Scan Context [40]	32.91	22.89	75.96	65.06	66.40	53.69	50.90	44.57
	M2DP [33]	34.69	23.14	45.03	32.41	44.62	34.34	39.34	32.95
NetVLAD-based	PointNetVLAD [69]	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
	E2PN-NetVLAD	**91.83**	**82.46**	**88.04**	**84.73**	80.58	**80.93**	**79.84**	**79.31**
GeM-based	MinkLoc3D [42]	**97.91**	**93.76**	95.04	86.01	**91.19**	81.17	**88.48**	82.66
	E2PN-GeM	93.24	84.79	**95.29**	**88.08**	90.46	**83.67**	87.68	**83.29**

Velodyne HDL-64E, randomly downsampled to 4096 points, and the points of ground are not removed. The point clouds are scaled to [-1, 1] with zero mean. Sequence 00 is chosen for evaluation because of its large set of pairs for loop closure in the same orientation. While sequence 08 is chosen for evaluation because it contains 100% reverse loop closure. Reverse loop closures provide scenes with 180-degree viewing angle differences and advance 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. For sequence 08, the first 85 seconds and segment from 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 3.2 reports the average recall at the top 1% and top 1 for place recognition in sequence 00 and 08. All methods are trained using the same Oxford training data set. The table shows that the SE(3)-invariant property in the proposed methods helps them perform better in these challenging scenarios, supporting the better generalization claim.

Table 3.2: Average recall (%) at top 1% on KITTI sequence 00 and 08. * is with attentive downsampling.

	KITTI Sequence 00		KITTI Sequence 08	
	AR@1%	AR@1	AR@1%	AR@1
PointNetVLAD [69]	73.18	70.68	32.47	17.61
EPN-NetVLAD* (Ours)	78.21	61.90	63.84	37.69
E2PN-NetVLAD (Ours)	**84.96**	**71.67**	**71.38**	**57.23**
MinkLoc3D [42]	28.07	4.01	17.30	3.50
E2PN-GeM (Ours)	**86.22**	**77.94**	**71.70**	**59.75**

Table 3.3: 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 on Oxford benchmark.

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

3.3.6 Attentive Downsampling

We design an experiment to test the performance of point cloud downsampling using an attention mechanism. We study EPN-NetVLAD'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 test set includes point clouds in 23 sequences of the Oxford benchmark. The results of different downsamplings are presented in Table 3.3. It shows that an attention mechanism to downsample point clouds can maintain high place recognition performance up to 50% downsampling rate.

Table 3.4: Data augmentation experiment on Oxford benchmark with different training data size and whether random transformation is applied during training. * is with attentive downsampling.

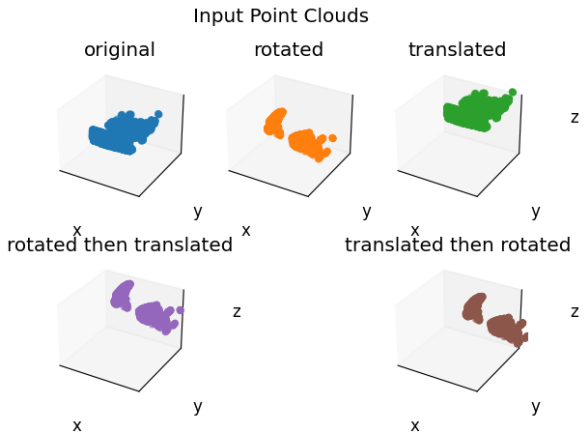
	Random Transformation	Training Size: 3 Sequences AR@1%	Training Size: 3 Sequences AR@1	Training Size: 45 Sequences AR@1%	Training Size: 45 Sequences AR@1
PointNetVLAD [69]		69.38	54.00	86.88	73.12
PointNetVLAD [69]	✓	80.85	65.55	84.94	71.39
EPN-NetVLAD * (Ours)		75.15	57.51	89.17	77.69
E2PN-NetVLAD (Ours)		85.16	70.61	93.78	85.04
MinkLoc3D [42]	✓	-	-	97.91	93.76
E2PN-GeM (Ours)		88.49	76.73	94.76	87.45

3.3.7 Data Augmentation Experiment

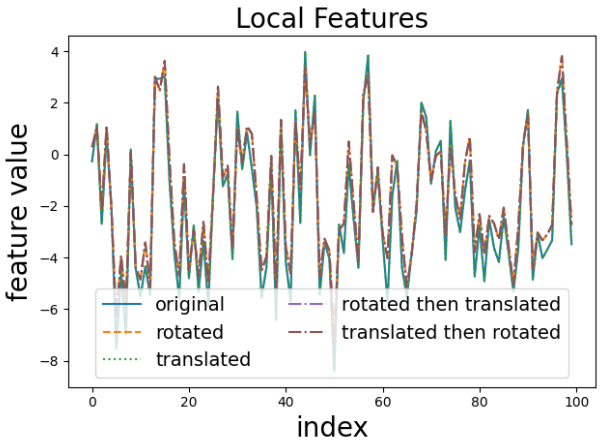
In Table 3.4, we experiment with different amounts of training data in the Oxford benchmark. PointNetVLAD relies on both random transformation and increasing training data size to achieve high performance, whereas E2PN-NetVLAD can achieve similar performance with only training on three sequences. MinkLoc3D performs the best among all methods. However, it still requires random transformations in the training data.

3.3.8 Descriptors with SE(3)-Invariant Property

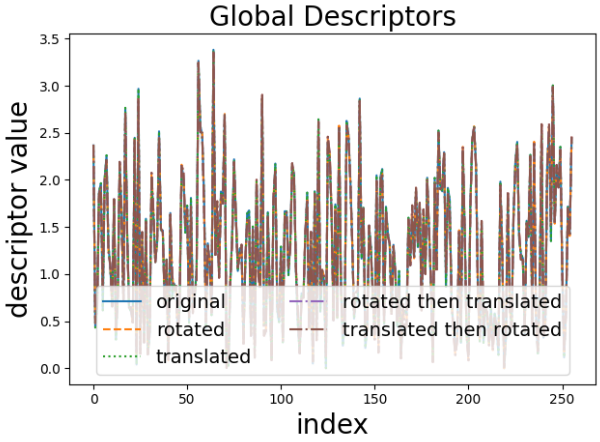
In addition to the place recognition experiments on Oxford and in-house benchmark, we take one point cloud in the Oxford benchmark, and construct simulated data by manually rotating and translating the point cloud to test the model performance with severe rotation and translation. We visualize the local features and global descriptors when the input point cloud is transformed under rotation, translation, rotated, then translated, and translated, then rotated. Figure 3.3 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.



(a) point clouds under transformations



(b) SE(3)-invariant local features (first 100)



(c) SE(3)-invariant global features

Figure 3.3: Visualization of one input point cloud, the same point cloud under transformations, their corresponding local features, and the output global descriptors. There are 1024 local features for each point in the point cloud. We only show the first 100 features of the first point for this visualization.

Table 3.5: Average recall (%) at top 1% for performing place recognition task on simulated data where the point clouds are transformed under rotation or/and translation. * is with attentive downsampling.

Method	Rotation	Translation	Rotation and Translation
PointNetVLAD	6.60	3.21	2.96
MinkLoc3D	13.71	**100.00**	13.77
EPN-NetVLAD*	30.13	40.69	27.55
EPN-NetVLAD	49.62	**100.00**	51.19
E2PN-NetVLAD	71.38	**100.00**	68.81
E2PN-GeM	**75.97**	**100.00**	**73.91**

3.3.9 Oxford Experiment with Random SE(3) Transformations

We augment point clouds from the Oxford data set with random SE(3) transformations to create a more challenging place recognition experiment. The point clouds 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 (uniformly sampled between $-\pi$ and π along all three axes) using SciPy library [71], and 3D translations are applied with a standard deviation of 1.0. We use the model trained on 45 sequences of the Oxford benchmark to perform place recognition on this augmented data set which includes a total of 1320 point clouds that has 440 query point clouds, each of which has two positive pairs. Table 3.5 shows the average recall rate. We observe that with the attention downsampling module, the performance of EPN-NetVLAD decreases. The attentive downsampling module selects different points to feed into the network, whereas other methods compute over the same points in this case.

3.3.10 Run Time Performance

We tested our method on a system equipped with an Intel i9-10900K CPU and an Nvidia GeForce RTX 3090. For obtaining one global descriptor from one

Table 3.6: Run time performance per point cloud which contains 4096 points. The run times are computed without any network optimization. * attentive downsampling is performed to reduce point cloud size to 2048 points.

	Parameters	Run Time (s)
PointNetVLAD [69]	19,779,145	0.006
MinkLoc3D [42]	1,055,713	0.005
EPN-NetVLAD (Ours)*	17,135,376	2.052
E2PN-NetVLAD (Ours)	17,167,488	0.079
E2PN-GeM (Ours)	192,513	0.082

3D point cloud with 4096 points, we record the number of parameters in the network and report the run time performance per point cloud in Table 3.6. E2PN-GeM has the lowest number of parameters. MinkLoc3D has the shortest inference time. 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 E2PN are coded with custom functions to perform group convolutions, while other networks have network structures optimized on GPU. Thus, it is possible that SE(3)-equivariant networks can be further optimized to improve run time.

3.4 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.

3.5 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.

Chapter 4

SE3ET: SE(3)-Equivariant Transformer for Low-Overlap Point Cloud Registration

In this chapter, we introduce SE3ET, an SE(3)-equivariant registration framework that employs equivariant point convolution and equivariant transformer design to learn expressive and robust geometric features. Partial point cloud registration is a challenging problem in robotics, especially when the robot undergoes a large transformation, causing a significant initial pose error and a low overlap between measurements. This work proposes exploiting equivariant learning from 3D point clouds to improve registration robustness. We propose SE3ET, an SE(3)-equivariant registration framework that employs equivariant point convolution and equivariant transformer designs to learn expressive and robust geometric features. We tested the proposed registration method on indoor and outdoor benchmarks where the point clouds are under arbitrary transformations and low overlapping ratios. We also provide generalization tests and run-time performance. The open-source software is available at
`https://github.com/UMich-CURLY/SE3ET`.

4.1 Introduction

has gained significant attention recently due to advancements in 3D sensor technology and computational resources. It seeks to determine the optimal transformation

between two point clouds, addressing core challenges in computer vision, computer graphics, and robotics [17, 94]. These tasks include 3D localization, 3D reconstruction, pose estimation, and simultaneous localization and mapping (SLAM) [36].

Partial-to-partial registration is widespread yet challenging in robotics applications. Many point cloud registration methods require sufficient overlap between two point clouds to find an accurate transformation [76]. Recent works such as Predator [34] and GeoTransformer [63] focus on low-overlap point cloud registration. However, these methods are not optimized for cases with significant initial pose errors, which are common in robotics. The current state-of-the-art works' limitations are shown in Figure 4.1.

Recent progress in equivariant networks for 3D point clouds [19, 14, 101, 49] enables neural networks to learn geometric features that preserve transformation information, providing a robust method to handle arbitrary transformations. We propose leveraging this advancement to improve point cloud registration solutions.

Few existing learning-based methods consider arbitrary transformations in the network architecture. For example, YOHO [72] applies equivariant feature learning using icosahedral-group convolution to learn rotation-equivariant point features. However, the framework can be further optimized for low-overlap registration.

In this work, we propose SE3ET, a more robust and efficient SE(3)-equivariant low-overlap point cloud registration framework. This framework (shown in Figure 4.2) includes an equivariant feature learning encoder-decoder and an equivariant transformer module, fully leveraging the advantages of equivariant networks. Our main contributions are:

1. Equivariant convolutions and transformers improve robustness to point clouds with low overlap and large pose changes.

2. We propose four designs of equivariant transformers, each offering unique benefits.

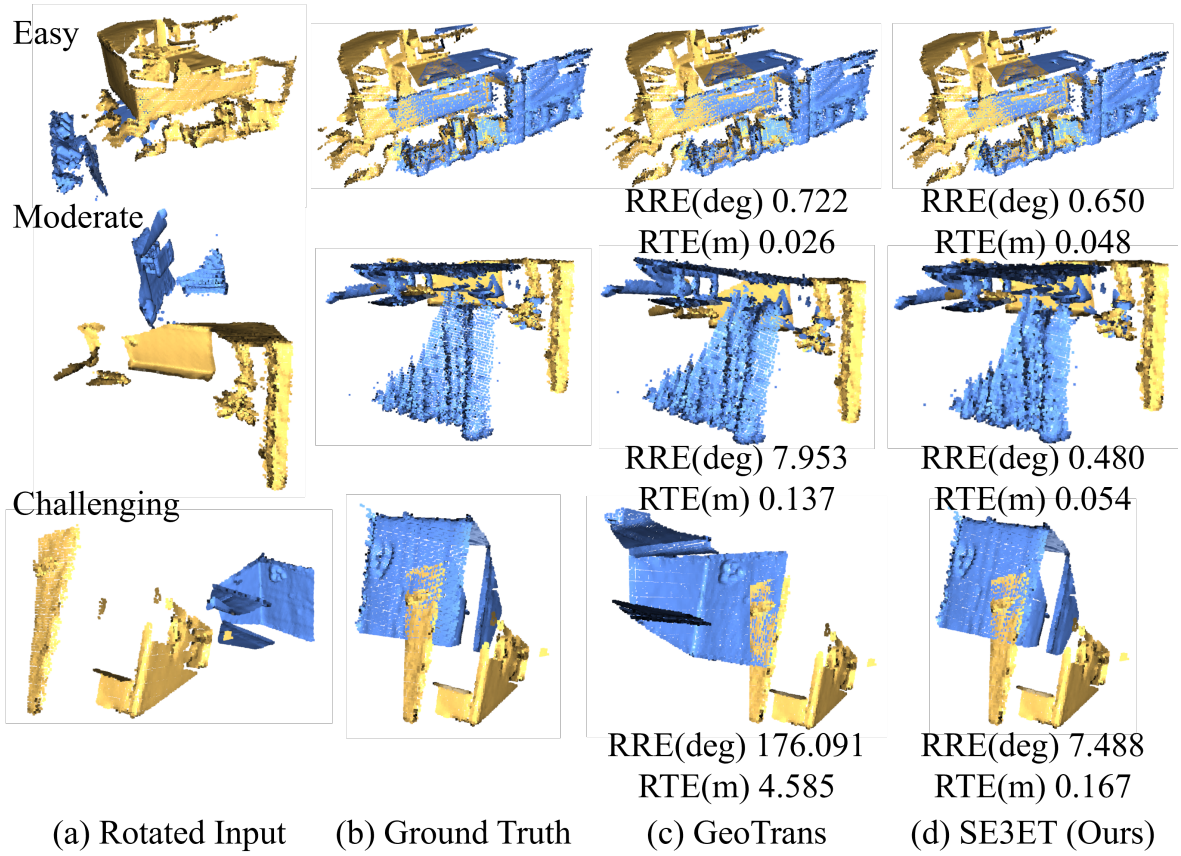


Figure 4.1: SE3ET can register two low-overlap point clouds with significant rotations and translations. This qualitative result is performed on rotated 3DLoMatch, where the first row is an easy example (28.72 % overlapping ratio with multiple overlapping surfaces), the middle row is a moderate example (10.51 % overlapping ratio with multiple overlapping surfaces), and the last row is a challenging example (26.75 % overlapping ratio with only one overlapping surface).

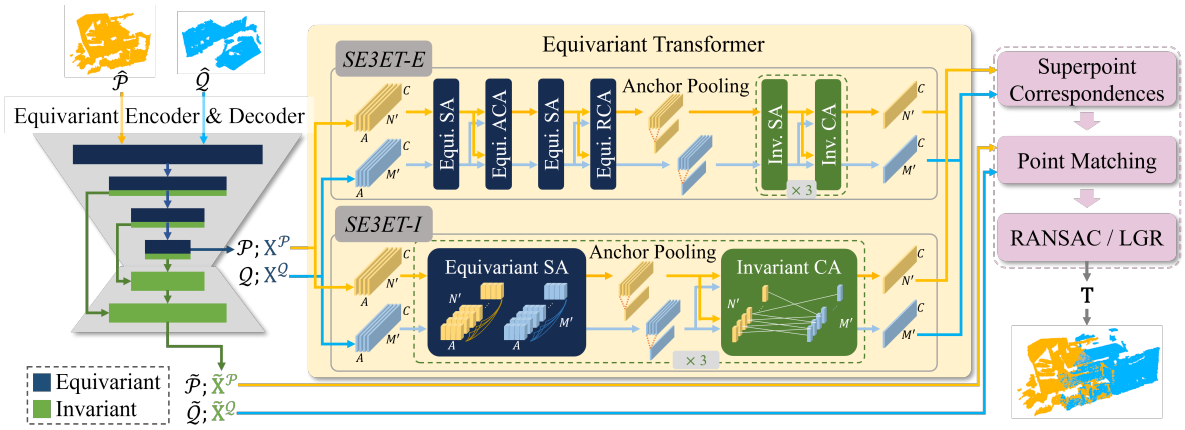


Figure 4.2: The proposed point cloud registration framework includes a SE(3)-equivariant feature encoder and decoder and an equivariant transformer design for learning the point correspondences of superpoints. The dark blue blocks and arrows are equivariant, and the green blocks and arrows are invariant. We propose two transformers structures *SE3ET-E* and *SE3ET-I*. Here, SA stands for self-attention, and CA stands for cross-attention.

3. We leverage the octahedral rotation group to improve efficiency and performance. This approach can also accommodate voxel downsampling.

4. Open-source software is available at `https://github.com/UMich-CURLY/SE3ET`.

4.2 Preliminary

We provide preliminaries on E2PN architecture of learning SE(3)-equivariant features.

4.2.1 Equivariance and Invariance

Invariance is where the output of a function remains unchanged under transformations of the input, expressed as $f_{inv}(g \circ x) = f_{inv}(x)$. This is useful for tasks like classification, where different input transformations represent the same object.

Equivariance is a more generalized form where the function's output undergoes the same transformation as the input, expressed as $f_{equ}(g \circ x) = g \circ f_{equ}(x)$. Equivariant

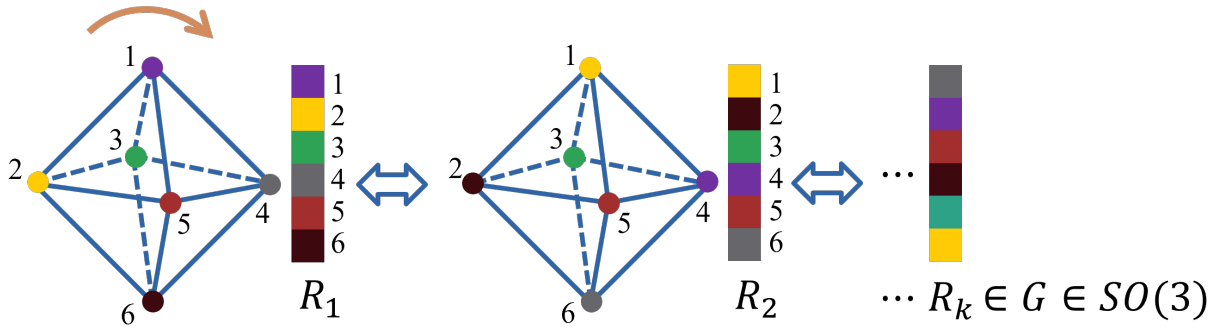


Figure 4.3: The geometric (octahedron shape) and algebraic (color bricks) illustration of using permutation to recover discretized rotation group. Each color represents the feature of one anchor (vertex), and the different order of the combination of features represents the discretized rotation defined in the network. If the octahedron is rotated 90 degrees clockwise, the order of the features in the anchor dimension changes accordingly. The discretization of the rotation groups is derived from the permutation of the discrete anchors. For $A = 6$, the rotation group contains 24 rotations.

representations are more expressive than invariant representations, offering better cross-domain generalizability.

4.2.2 E2PN

In the SE3ET framework, E2PN [101] is used in the encoder and decoder modules to learn equivariant and invariant local features from 3D point clouds. E2PN extends Kernel Point Convolutions (KPConv) [68] by discretizing SO(3) into a polyhedral rotation group G, approximating SE(3) as $\mathbb{R}^3 \times G$. Feature maps are defined on $\mathbb{R}^3 \times V$, with V representing polyhedral vertices, or *anchors*. These maps are computed via quotient-space convolution. E2PN features have an additional anchor dimension. When a discretized rotation group transforms the point cloud, the features are permuted on the anchor dimension, as shown in Figure 4.3. Conventional transformers are not optimal for E2PN features, motivating us to develop an equivariant transformer.

4.3 Methodology

Point cloud registration computes a (rigid) transformation $\mathbf{T} = \{(\mathbf{R}, \mathbf{t}) \mid \mathbf{R} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3\} \in \mathrm{SE}(3)$ that aligns two given partially overlapped point clouds $\hat{\mathcal{P}} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ and $\hat{\mathcal{Q}} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \ldots, M\}$.

The proposed framework, shown in Figure 4.2, consists of three main components. Initially, an SE(3)-equivariant feature encoder (Section 4.3.1) is employed to extract equivariant and invariant features from input point clouds at various resolutions. Subsequently, coarse-level features from two point clouds learn correlations with each other in an equivariant transformer (Section 4.3.2), comprising interleaved self-attention and cross-attention modules for spatial and rotational feature enhancement within and between point clouds. Finally, both coarse and fine-level SE(3)-invariant features are utilized for point matching in the registration module (Section 4.3.4).

4.3.1 Equivariant Feature Encoder and Decoder

We adopt GeoTransformer's multi-stage encoder-decoder structure [63], following a Feature Pyramid Network [48] to extract multi-scale features from downsampled point clouds. Different from conventional approaches, we employ E2PN [101], an SE(3)-equivariant point convolutional network, for SE(3)-equivariant convolution. Point clouds downscaled to the coarsest level (termed "superpoints") are denoted as $\mathcal{P} \in \mathbb{R}^{N' \times 3}$, $\mathcal{Q} \in \mathbb{R}^{M' \times 3}$, with their equivariant features $\mathbf{X}^{\mathcal{P}} \in \mathbb{R}^{N' \times A \times C}$, $\mathbf{X}^{\mathcal{Q}} \in \mathbb{R}^{M' \times A \times C}$, where N', M' denote the downsampled points, C represents the feature channels, and $A = |V|$ denotes the anchor size. These equivariant features undergo enhancement in an equivariant transformer. After each encoder stage, max-pooling along the anchor dimension yields SE(3)-invariant features. These features are then passed into the decoder, facilitating the extraction of SE(3)-invariant features $\widetilde{\mathbf{X}}^{\mathcal{P}}$ and $\widetilde{\mathbf{X}}^{\mathcal{Q}} \in \mathbb{R}^{N' \times C}$ for fine-level point clouds $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$, essential for fine point matching (Section 4.3.4).

Different from E2PN's icosahedral implementation ($A = 12$), we implement the octahedral ($A = 6$) finite rotation groups for more efficient computation. By design, we can express each rotation in the discretized rotation group G using a permutation of the vertices V. A geometric illustration is in Figure 4.3.

4.3.2 Equivariant Transformer Design

The original transformer [70] operates using dot products of two matrices. However, the equivariant features derived from the encoder are 3-dimensional tensors (comprising point, anchor, and feature dimensions), each with its own geometric representation. Flattening these tensors would disregard their intrinsic structure. We propose equivariant transformer designs that preserve the additional anchor dimension to address this. We propose three unique equivariant transformer designs, each crafted for different purposes: (1) learning attention weights on the point dimension while accounting for anchor features, (2) learning attention weights on the anchor dimension, and (3) learning attention weights for each discretized rotation group. Additionally, we propose an invariant transformer design that learns attention weights on the point dimension when the input features are invariant, ensuring stability to SE(3) transformations.

We propose equivariant self-attention and cross-attention modules to enhance features of superpoints by gathering information across various spatial locations and orientations. Self-attention modules facilitate feature interaction within a point cloud, while cross-attention modules enable feature communication between pairs of point clouds. Below, we detail each module design.

4.3.2.1 Equivariant Self-Attention (ESA) Module

We introduce an equivariant self-attention module, extending self-attention methods in [63] while ensuring equivariance to SE(3) transformations. This module allows

consistent behavior under rigid body transformations.

We will use the lower subscript to denote the feature after a certain layer for simplicity of notation. For example, $\mathbf{x}_{\texttt{SA}}$ is the feature acquired from the self-attention module.

Since the self-attention module is conducted per point cloud, we focus on operations within point cloud $\mathcal{P}$, which similarly apply to $\mathcal{Q}$. Equivariant superpoint features $\mathbf{X}^{\mathcal{P}}$ serve as query, key, and value inputs. We follow [63] to use geometric information to provide geometric structure embedding $\mathbf{P}^{\mathcal{P}} \in \mathbb{R}^{N' \times N' \times C}$.

For a point indexed i in $\mathcal{P}$ at anchor coordinate r, attention between such elements is computed using trainable weight matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ and $\mathbf{W}^P \in \mathbb{R}^{C \times C}$. Output features $\mathbf{x}^{\mathcal{P}}_{\texttt{SA},ir}$ are obtained via weighted summation over points. A subsequent feed-forward layer, as in [70], refines learned features, resulting in $\mathbf{X}^{\mathcal{P}}_{\texttt{SA}} \in \mathbb{R}^{N' \times A \times C}$.

$$\mathbf{a}_{\texttt{SA},ir,jr} = \frac{(\mathbf{x}_{ir}\mathbf{W}^Q)(\mathbf{x}_{jr}\mathbf{W}^K + \mathbf{p}_{i,j}\mathbf{W}^P)^\intercal}{\sqrt{C}}, \tag{4.1}$$

$$\mathbf{x}^{\mathcal{P}}_{\texttt{SA},ir} = \sum_{j=1}^{N'} \text{Softmax}_j(\mathbf{a}_{\texttt{SA},ir,jr})\mathbf{x}_{jr}\mathbf{W}^V, \tag{4.2}$$

Compared with conventional self-attention among the point features, our equivariant self-attention module allows different attention values at different anchor coordinates for the same pair of points, similar to multi-head self-attention, but with the equivariant property.

4.3.2.2 Invariant Cross-Attention (ICA) Module

The cross-attention mechanism integrates two separate inputs, typically observed in point cloud registration tasks with paired point clouds $\mathcal{P}$ and $\mathcal{Q}$. We focus on operations within $\mathcal{P}$; analogous operations apply to $\mathcal{Q}$.

In the ICA module, attention is conducted on SE(3)-invariant features derived from equivariant self-attention features $\mathbf{X}^{\mathcal{P}}_{\texttt{SA}}$ and $\mathbf{X}^{\mathcal{Q}}_{\texttt{SA}}$ for $\mathcal{P}$ and $\mathcal{Q}$, respectively.

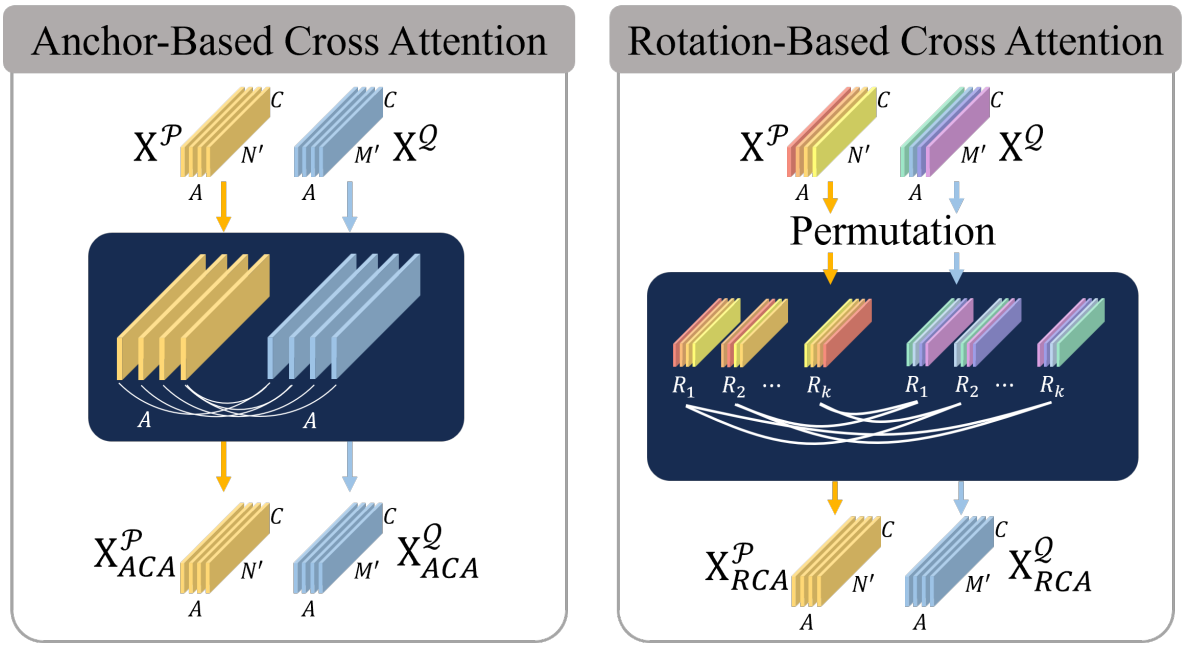


Figure 4.4: An illustration of the equivariant anchor-based cross-attention (ACA) and rotation-based cross-attention modules (RCA).

Pooling on the anchor dimension yields invariant features $\mathbf{X}^{\mathcal{P}}_{\texttt{SA-inv}} \in \mathbb{R}^{N' \times C}$ and $\mathbf{X}^{\mathcal{Q}}_{\texttt{SA-inv}} \in \mathbb{R}^{M' \times C}$.

In the ICA module, features $\mathbf{X}^{\mathcal{P}}_{\texttt{SA-inv}}$ serve as queries, and $\mathbf{X}^{\mathcal{Q}}_{\texttt{SA-inv}}$ as keys. The attention value between point i in $\mathcal{P}$ and point j in $\mathcal{Q}$ is computed using trainable weight matrices.

$$\mathbf{a}_{\texttt{ICA},i,j} = \frac{(\mathbf{x}^{\mathcal{P}}_{\texttt{SA-inv},i}\mathbf{W}^{Q})(\mathbf{x}^{\mathcal{Q}}_{\texttt{SA-inv},j}\mathbf{W}^{K})^{\mathsf{T}}}{\sqrt{C}} \tag{4.3}$$

Depending on the desired output—equivariant or invariant—the values can be equivariant features $\mathbf{X}^{\mathcal{Q}}_{\texttt{SA}}$ or invariant features $\mathbf{X}^{\mathcal{Q}}_{\texttt{SA-inv}}$. The collection of equivariant output features is denoted as $\mathbf{X}^{\mathcal{P}}_{\texttt{ICA}} \in \mathbb{R}^{N' \times A \times C}$, while invariant output features are denoted as $\mathbf{X}^{\mathcal{P}}_{\texttt{ICA-inv}} \in \mathbb{R}^{N' \times C}$. Take $\mathbf{X}^{\mathcal{P}}_{\texttt{ICA},i}$ for point i in $\mathcal{P}$ as an example,

$$\mathbf{x}^{\mathcal{P}}_{\texttt{ICA},i} = \sum_{j=1}^{M'} \text{Softmax}_j(\mathbf{a}_{\texttt{ICA},i,j})\mathbf{x}^{\mathcal{Q}}_{\texttt{SA},j}\mathbf{W}^{V} \tag{4.4}$$

4.3.2.3 Equivariant Anchor-Based Cross-Attention (ACA) Module

In this module, we learn cross-attention scores for each anchor dimension. For point cloud $\mathcal{P}$, its equivariant feature from the self-attention module $\mathbf{X}_{\texttt{SA}}^{\mathcal{P}}$ serves as query and $\mathbf{X}_{\texttt{SA}}^{\mathcal{Q}}$ serves as key and value. Attention computation between two points in $\mathcal{P}$ and $\mathcal{Q}$ is performed via trainable weight matrices.

$$\mathbf{a}_{\texttt{ACA_raw},ir,js} = \frac{(\mathbf{x}_{\texttt{SA},ir}^{\mathcal{P}}\mathbf{W}^{Q})(\mathbf{x}_{\texttt{SA},js}^{\mathcal{Q}}\mathbf{W}^{K})^{\mathsf{T}}}{\sqrt{C}} \tag{4.5}$$

Normalization is applied on both anchor and spatial dimensions to stabilize feature learning. Softplus ensures the non-negativity of raw attention, preventing false point-matching from distracting the anchor-wise attention between two point clouds. Global anchor-based attention, denoted as $\mathbf{a}_{\texttt{ACA_anchor}}$, captures correlations between anchors across all points by average pooling on the point dimension.

$$\mathbf{a}_{\texttt{ACA_anchor},r,s} = \frac{1}{N'M'}\sum_{i=1}^{N'}\sum_{j=1}^{M'}\text{Softplus}(\mathbf{a}_{\texttt{ACA_raw},ir,js}) \tag{4.6}$$

We conduct normalization in both the anchor and spatial dimensions for stable feature learning. Anchor-wise, we calculate the normalized global anchor attention:

$$\mathbf{a}_{\texttt{ACA_norm_anchor},r,s} = \frac{\mathbf{a}_{\texttt{ACA_anchor},r,s}}{\sum_{s=1}^{A}\mathbf{a}_{\texttt{ACA_anchor},r,s}} \tag{4.7}$$

Spatial-wise, we apply softmax on the j dimension to normalize the spatial dimension.

$$\mathbf{a}_{\texttt{ACA_norm_spatial},ir,js} = \text{Softmax}_j(\mathbf{a}_{\texttt{ACA_raw},ir,js}) \tag{4.8}$$

We multiply Equation (4.7) and Equation (4.8) to obtain the resulting attention for

both the spatial and anchor dimensions.

$$\mathbf{a}_{\texttt{ACA},ir,js} = \mathbf{a}_{\texttt{ACA_norm_spatial},ir,js}\mathbf{a}_{\texttt{ACA_norm_anchor},r,s} \tag{4.9}$$

Output features for each point in $\mathcal{P}$ are obtained through weighted summation over points in $\mathcal{Q}$, followed by a feed-forward layer.

$$\mathbf{x}^{\mathcal{P}}_{\texttt{ACA},ir} = \sum_{j=1}^{M'}\sum_{s=1}^{A}\mathbf{a}_{\texttt{ACA},ir,js}\mathbf{x}^{\mathcal{Q}}_{\texttt{SA},js}\mathbf{W}^{V}, \tag{4.10}$$

ACA preserves equivariance, as the attention value depends solely on feature contents, regardless of anchor indices. Figure 4.4 shows the illustration. This design ensures equivariant preservation, which is crucial for consistent behavior under rotations.

4.3.2.4 Equivariant Rotation-Based Cross-Attention (RCA) Module

In this version of cross-attention, we learn the cross-attention scores for each discretized rotation in the rotation group. Figure 4.4 shows the illustration.

First, we use the permutation layer from E2PN [101], which is mentioned in Section 4.2.2 and Section 4.3.1, to reconstruct feature maps defined on the discretized rotation group G from the features defined on anchors V. We denote the permuted feature corresponding to the rotation $g \in G$ as:

$$\mathbf{x}^{\mathcal{Q}}_{\texttt{Permute},jg} = \text{Permute}_g(\{\mathbf{x}^{\mathcal{Q}}_{\texttt{SA},js}\}_{s=1,\dots,A}) \tag{4.11}$$

After obtaining the feature corresponding to the rotation groups, the raw attention between two input features can be computed as:

$$\mathbf{a}_{\texttt{RCA_raw},i,jg} = \frac{(\mathbf{x}^{\mathcal{P}}_{\texttt{SA},i}\mathbf{W}^{Q})(\mathbf{x}^{\mathcal{Q}}_{\texttt{Permute},jg}\mathbf{W}^{K})^{\mathsf{T}}}{\sqrt{C}} \tag{4.12}$$

We carry out normalization in both the rotation and spatial dimensions for consistent feature learning.

$$\mathbf{a}_{\texttt{RCA_rot},g} = \frac{1}{N'M'} \sum_{i=1}^{N'} \sum_{j=1}^{M'} \text{Softplus}(\mathbf{a}_{\texttt{RCA_raw},i,jg}) \tag{4.13}$$

$$\mathbf{a}_{\texttt{RCA_norm_rot},g} = \frac{\mathbf{a}_{\texttt{RCA_rot},g}}{\sum_{g=1}^{|P|} \mathbf{a}_{\texttt{RCA_rot},g}} \tag{4.14}$$

$$\mathbf{a}_{\texttt{RCA_norm_spatial},i,jg} = \text{Softmax}_j(\mathbf{a}_{\texttt{RCA_raw},i,jg}) \tag{4.15}$$

We multiply Equation (4.14) and Equation (4.15) to obtain attention for both the spatial and rotation dimensions.

$$\mathbf{a}_{\texttt{RCA},i,jg} = \mathbf{a}_{\texttt{RCA_norm_spatial},i,jg} \quad \mathbf{a}_{\texttt{RCA_norm_rot},g} \tag{4.16}$$

The output feature for the i'th point in $\mathcal{P}$, denoted as $\mathbf{x}^{\mathcal{P}}_{\texttt{RCA},i} \in \mathbb{R}^{A \times C}$, can be written as:

$$\mathbf{x}^{\mathcal{P}}_{\texttt{RCA},i} = \sum_{g=1}^{|P|} \sum_{j=1}^{M'} \mathbf{a}_{\texttt{RCA},i,jg} (\mathbf{x}^{\mathcal{Q}}_{\texttt{Permute},jg} \mathbf{W}^V) \tag{4.17}$$

After passing through the feed-forward layer, we denote the collection of the output features as $\mathbf{X}^{\mathcal{P}}_{\texttt{RCA}} \in \mathbb{R}^{N' \times A \times C}$.

4.3.3 Equivariant Transformer Configurations

To integrate the proposed transformer modules for point cloud registration, we present two configurations: the equivariant version, named *SE3ET-E*, with all the proposed modules, and *SE3ET-I*, which employs invariant cross-attention modules. Their structures are illustrated in Figure 4.2.

SE3ET-E includes all the previously introduced modules, leveraging the full potential of equivariant features and optimizing performance where preserving geometric structure is crucial.

SE3ET-I combines equivariant self-attention with invariant cross-attention modules,

reducing network complexity while maintaining SE(3)-invariant attention learning to enhance registration robustness under arbitrary transformations. It is suitable for scenarios with large point clouds.

These configurations address different challenges: SE3ET-E for high geometric fidelity and SE3ET-I for robust transformation invariance and scalability.

4.3.4 Registration and Estimate Transformation

We adopt the methodology of GeoTransformer [63], which consists of the following steps: superpoint-matching, fine point-matching, and a combined overlap-aware circle loss and point-matching loss for self-supervised learning. The output features from the transformer module are fed into the superpoint-matching module to identify patch correspondence. The fine-point matching module finds direct correspondence for a denser set of points. Afterward, transformation estimation is conducted either using RANSAC [26] or a local-to-global (LGR) approach [63]. When utilizing RANSAC, a certain number of samples are chosen based on the high correspondence score derived from the fine-matching module. Lastly, the transformation between the two input point clouds is calculated via the Open3D library [99], considering the paired points.

4.4 Experimental Results

To show the generalizability of the proposed work, we conducted experiments on both indoor 3DMatch/3DLoMatch benchmark (Section 4.4.1) and outdoor KITTI point clouds (Section 4.4.2). We also tested the algorithm's generalizability (Section 4.4.3) and run-time performance (Section 4.4.4).

4.4.1 Indoor Benchmark: 3DMatch/3DLoMatch

3DMatch [92] is a data set that provides noisy and partial RGB-D scanning data for indoor reconstruction. It consists of 62 scenes of point clouds, of which 46 are for training, 8 are for validation, and the other 8 scenes are for testing. 3DMatch contains point cloud pairs with over 30% of overlapping rate. Predator [34] extended it and created 3DLoMatch that only contains point cloud pairs with 10% to 30% overlap ratio for evaluating low overlapping point clouds.

To show the proposed method's robustness under arbitrary pose changes, following YOHO [72], we also report results where the point clouds in the testing set are transformed with *random rotations* since the original data only contains limited pose changes.

4.4.1.1 Implementation Details

We use the octahedral rotation group with $A = 6$. There are four stages in the encoder, and the equivariant features output from the encoder have feature size $C = 1024$. SE3ET-E and SE3ET-I models are trained for 40 epochs with an initial learning rate of 10^{-4}.

4.4.1.2 Evaluation Metrics

Following the literature [4, 34, 63], we report *Registration Recall (RR)*, as the registration metric. Given the estimated relative pose between a pair of point clouds, the RMSE of the locations of ground truth corresponding points between the aligned point clouds is calculated. RR is the percentage of point cloud pairs with RMSE smaller than a threshold (0.2 meters). This metric reflects the overall accuracy of the registration.

For correspondence matching, we report *Inlier Ratio (IR)*, defined as the proportion of estimated corresponding pairs of points of which the distance is smaller than a

threshold (0.1 meters) under the ground truth transformation. This metric indicates the effectiveness of correspondence matching. A higher IR suggests a more precise match of correspondence between two point clouds.

We also report *Feature Matching Recall (FMR)*, calculating the percentage of point cloud pairs with an inlier ratio higher than a certain threshold (5%). This metric gauges how well the method can recover the pose with high confidence. A higher FMR reflects a more effective correspondence-based registration process, assuming the outlier matchings can be filtered out during the transformation estimation process.

To better understand the method's performance, we evaluate using the following absolute metrics. *Relative Translation Error (RTE)*: the Euclidean distance between the estimated translation and the ground truth translation vector. *Relative Rotation Error (RRE)*: an isotropic error between the estimated rotation matrix $\hat{\mathbf{R}}$ and the ground truth rotation matrix $\mathbf{R}$, calculated as $\arccos(\frac{\mathrm{tr}(\hat{\mathbf{R}}^\intercal \mathbf{R}-1)}{2})$.

4.4.1.3 Evaluation Results

We compare the original and rotated 3DMatch/3DLoMatch result with state-of-the-art point cloud registration methods in Table 4.1. We reproduce GeoTransformer's [63], PEAL's [90], and CoFiNet's [88] performance with their pre-trained weight. We report FCGF [16], D3Feat [4], Predator [34], SpinNet [2], and YOHO [72] result from [72], and compare RoReg's [73], BUFFER's [1], and RoITr's [89] performance according to their paper. The registration is performed with 5000 sample points and 50k iterations for RANSAC except YOHO and RoReg, which only run for 1k iterations, as stated in their paper.

From Table 4.1, our method performs the best among all other methods in arbitrary rotation, regardless of the overlapping ratio, showing superior robustness to low overlap and arbitrary rotations. Although RoReg demonstrates slightly better performance on the original 3DMatch, it fails to sustain the advantage on 3DLoMatch.

We exclude PEAL [90] in the ranking because PEAL uses overlap prior, which requires estimated transformation as input by running GeoTransformer beforehand. We also exclude RoITr [89] in the ranking because it requires normal directions as input. When comparing these two methods that require extra information, despite our lower inlier ratio, which is an intermediate metric, we maintain competitive results on other metrics, including the final registration recall.

We further examine the result for using different numbers of samples for the original 3DMatch/3DLoMatch in Table 4.2 and rotated 3DMatch/3DLoMatch Table 4.3. In Table 4.2, corresponding to the not-rotated testing set, our models yield the best performance in the 3DLoMatch testing set, showing the robust performance of our method in harsh low-overlapping conditions. In the 3DMatch testing set, where the overlap ratio is higher, we observe mixed results in the inlier ratio (IR). We hypothesize that local point cloud patches with rotational symmetry could obscure the rotation-invariant features. Because the not-rotated testing set only includes very limited relative pose changes between a point cloud pair, the non-equivariant baseline methods will not suffer from the pose ambiguity and can overfit the feature descriptor to the local pose difference between the local patches. Compared with other equivariant registration methods (YOHO [72] and RoReg [73]), our method performs better in most metrics. RoReg has slightly higher registration recalls in 3DMatch, but our method leads with a considerable margin in other metrics, especially in the 3DLoMatch benchmark.

From Table 4.3, it could be observed that our method offers better overall performance across different sample sizes when the input point clouds are subject to arbitrary rotations. Our method outperforms both the non-equivariant baseline (GeoTransformer [63]) and the equivariant baseline (YOHO [72]). Table 4.4 presents the 3DMatch and 3DLoMatch registration result in absolute metrics. Our method performs well under low overlap and arbitrary rotation.

Table 4.1: Evaluation on the original and rotated 3DMatch/3DLoMatch benchmark. Excluding the methods using extra information, the best result is shown in **bold**, and the second best is shown in the underline. SE3ET-E2 and SE3ET-I2 are the same as our proposed structures, with only half of the feature size.

Symmetry Category	Extra Information	Methods	3DMatch Original	3DMatch Rotated	3DLoMatch Original	3DLoMatch Rotated
Registration Recall (%) ↑						
None	✗	FCGF [16]	85.1	84.8	40.1	40.8
	✗	D3Feat [4]	81.6	83.0	37.2	36.1
	✗	Predator [34]	89.0	88.4	59.8	57.7
	✗	GeoTrans [63]	91.5	87.7	74.8	68.5
	✓	PEAL [90]	94.6	-	81.7	-
Invariant	✗	SpinNet [2]	88.6	88.4	59.8	58.1
	✓	RoITr [89]	91.9	94.7	74.8	77.2
Equivariant	✗	YOHO [72]	90.8	90.6	65.2	65.9
	✗	RoReg [73]	**92.9**	-	70.3	-
	✗	BUFFER [1]	**92.9**	-	71.8	-
	✗	*SE3ET-E/SE3ET-E2 (Ours)*	91.8/91.4	88.3/88.0	**77.0**/74.3	**72.0**/68.1
	✗	*SE3ET-I/SE3ET-I2 (Ours)*	92.6/91.8	**90.9**/87.8	75.9/74.9	70.8/70.1
Inlier Ratio (%) ↑						
None	✗	FCGF [16]	56.8	56.2	21.4	21.6
	✗	D3Feat [4]	39.0	39.2	13.2	13.5
	✗	Predator [34]	58.0	58.2	26.7	26.2
	✗	GeoTrans [63]	71.0	68.0	42.6	39.4
	✓	PEAL [90]	72.4	-	45.0	-
Invariant	✗	SpinNet [2]	47.5	47.2	20.5	20.1
	✓	RoITr [89]	82.6	82.3	54.3	53.2
Equivariant	✗	YOHO [72]	64.4	65.1	25.9	26.4
	✗	RoReg [73]	**81.6**	-	39.6	-
	✗	*SE3ET-E/SE3ET-E2 (Ours)*	72.5/73.4	**71.5**/71.0	44.3/**44.8**	**43.6**/42.4
	✗	*SE3ET-I/SE3ET-I2 (Ours)*	71.2/72.5	69.2/69.8	42.0/43.4	40.4/41.2
Feature Matching Recall (%) ↑						
None	✗	FCGF [16]	97.4	97.6	76.6	75.4
	✗	D3Feat [4]	95.6	95.5	67.3	67.6
	✗	Predator [34]	96.6	96.7	78.6	75.7
	✗	GeoTrans [63]	98.4	98.2	87.4	85.8
	✓	PEAL [90]	99.0	-	91.7	-
Invariant	✗	SpinNet [2]	97.6	67.5	75.3	75.3
	✓	RoITr [89]	98.0	98.2	89.6	89.4
Equivariant	✗	YOHO [72]	98.2	98.1	79.4	79.2
	✗	RoReg [73]	98.2	-	82.1	-
	✗	*SE3ET-E / SE3ET-E2 (Ours)*	98.5/97.9	98.2/98.0	**88.8**/86.8	**88.0**/85.8
	✗	*SE3ET-I / SE3ET-I2 (Ours)*	98.3/**98.6**	**98.5**/97.9	87.7/88.4	87.2/87.8

Table 4.2: Registration recall and inlier ratio evaluation results on 3DMatch and 3DLoMatch for different number of samples. RANSAC is run with 1k iterations for YOHO and RoReg and 50k iterations for other methods. * symbol means the method uses extra information, which is excluded from the ranking. The best result is in **bold**, and the second is in the underline.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Registration Recall (%) ↑										
Predator [34]	89.8	89.9	89.8	87.3	87.1	60.2	61.1	62.7	61.3	57.4
CoFiNet [88]	90.2	90.7	90.6	90.2	89.5	67.2	66.2	68.5	66.1	61.3
YOHO [72]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
RoReg [73]	**92.9**	**93.2**	**93.2**	**94.2**	**92.0**	70.3	70.2	68.3	67.6	64.9
GeoTrans [63]	91.5	91.3	90.4	90.9	90.1	74.8	74.4	74.7	73.7	73.2
BUFFER [1]	**92.9**	-	-	-	-	71.8	-	-	-	-
PEAL* [90]	94.6	93.7	93.7	93.9	93.4	81.7	81.2	80.8	80.4	80.1
RoITr* [89]	91.9	91.7	91.8	91.4	91.0	74.7	74.8	74.8	74.2	73.6
SE3ET-E (Ours)	91.8	91.9	91.1	91.5	90.4	**77.0**	**77.0**	**76.2**	**75.6**	**75.0**
SE3ET-I (Ours)	92.6	92.4	92.2	92.3	91.5	75.9	74.7	75.0	73.5	74.0
Inlier Ratio (%) ↑										
Predator [34]	58.0	58.4	57.1	54.1	49.3	26.7	28.0	28.3	25.7	25.7
CoFiNet [88]	49.9	51.2	51.9	52.1	52.2	24.4	25.9	26.7	26.8	27.1
YOHO [72]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
RoReg [73]	**81.6**	**80.7**	75.1	74.6	76.0	39.6	39.9	33.6	32.0	34.5
GeoTrans [63]	71.0	77.7	**82.3**	**84.1**	**85.3**	42.6	47.9	54.0	56.6	**58.4**
PEAL* [90]	72.4	79.1	84.1	86.1	87.3	45.0	50.9	57.4	60.3	62.2
RoITr* [89]	82.6	82.8	83.0	83.0	83.0	54.3	54.6	55.1	55.2	55.3
SE3ET-E (Ours)	72.5	77.8	81.2	82.5	83.2	**44.3**	**50.2**	**54.8**	**56.7**	57.9
SE3ET-I (Ours)	71.2	77.1	80.9	82.4	83.4	42.0	48.1	53.1	55.2	56.7
Feature Matching Recall (%) ↑										
Predator [34]	96.6	96.3	96.4	96.6	96.6	78.1	79.3	79.4	79.2	77.8
CoFiNet [88]	98.1	98.2	98.2	98.2	98.2	83.2	83.1	82.9	82.7	82.7
GeoTrans [63]	98.4	98.4	98.4	98.1	98.0	87.4	87.2	87.7	87.8	87.3
YOHO [72]	98.2	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
RoReg [73]	98.2	98.0	98.3	98.2	97.8	82.1	82.3	81.2	80.7	80.5
PEAL* [90]	99.0	99.0	99.1	99.1	98.8	91.7	92.4	92.5	92.9	92.7
RoITr* [89]	98.0	98.0	97.9	98.0	97.9	89.6	89.6	89.5	89.4	89.3
SE3ET-E (Ours)	**98.5**	**98.7**	**98.6**	**98.4**	**98.3**	**88.8**	**88.6**	**88.6**	**88.5**	**88.0**
SE3ET-I (Ours)	98.3	98.3	98.3	98.3	98.1	87.7	87.7	87.8	87.7	86.7

Table 4.3: Evaluation results on rotated 3DMatch and 3DLoMatch for different number of samples. RANSAC is run with 1k iterations for YOHO, as stated in their paper, and 50k iterations for other methods. The best result is shown in **bold**.

	Rotated 3DMatch					Rotated 3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Registration Recall (%) ↑										
YOHO [72]	90.6	-	-	-	-	65.9	-	-	-	-
GeoTrans [63]	87.7	88.2	88.4	87.5	87.9	68.5	67.8	68.0	66.7	66.5
SE3ET-E (Ours)	89.3	89.4	88.2	**89.3**	**89.1**	**72.0**	**71.1**	**71.3**	**70.6**	**69.7**
SE3ET-I (Ours)	**90.9**	**90.7**	**89.9**	89.1	88.5	70.8	70.3	71.0	69.8	68.1
Inlier Ratio (%) ↑										
YOHO [72]	65.1	-	-	-	-	26.4	-	-	-	-
GeoTrans [63]	68.0	74.8	80.1	82.3	83.6	39.4	44.1	50.7	53.6	55.5
SE3ET-E (Ours)	**71.5**	**77.6**	**81.7**	**83.3**	**84.4**	**43.6**	**49.5**	**55.1**	**57.5**	**59.0**
SE3ET-I (Ours)	69.2	75.7	80.1	81.9	83.1	40.4	46.2	51.8	54.4	56.2
Feature Matching Recall (%) ↑										
YOHO [72]	98.1	-	-	-	-	79.2	-	-	-	-
GeoTrans [63]	98.2	98.0	98.1	97.9	98.0	85.8	86.1	86.0	86.0	85.9
SE3ET-E (Ours)	98.2	**98.2**	98.1	98.0	98.1	**88.0**	**88.4**	**88.5**	**88.1**	**87.2**
SE3ET-I (Ours)	**98.5**	98.0	**98.2**	**98.4**	**98.2**	87.2	87.1	86.6	86.5	87.0

Table 4.4: Registration results on 3DMatch in absolute metrics Relative Translation Error (cm) and Relative Rotation Error (°). The best result is shown in **bold**.

Methods	3DMatch		Rotated 3DMatch		3DLoMatch		Rotated 3DLoMatch	
	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE
GeoTrans [63]	6.4	**1.825**	6.3	1.864	9.2	2.938	8.6	2.883
SE3ET-E (Ours)	6.4	1.893	**6.1**	**1.792**	9.0	2.973	**8.3**	**2.762**
SE3ET-I (Ours)	6.4	1.841	6.3	1.886	**8.8**	**2.914**	8.4	2.782

Table 4.5: Evaluation result on KITTI point clouds. The best result is shown in **bold**.

Method	RTE (cm) ↓	RRE (°) ↓	RR (%) ↑
FCGF [16]	9.5	0.30	96.6
D3Feat [4]	7.2	0.30	**99.8**
Predator [34]	6.8	0.27	**99.8**
CoFiNet [88]	8.2	0.41	**99.8**
SpinNet [2]	9.9	0.47	99.4
BUFFER [1]	5.4	0.22	-
GeoTrans-RANSAC-50k [63]	7.4	0.27	**99.8**
GeoTrans-LGR [63]	6.8	0.24	**99.8**
SE3ET-I-RANSAC50k (Ours)	6.3	0.25	99.7
SE3ET-I-LGR (Ours)	**4.7**	**0.21**	99.7

Table 4.6: Result for the generalization experiment. Evaluation results on KITTI point clouds with models trained on 3DMatch. The best result is shown in **bold**.

Method	RTE (cm) ↓	RRE (°) ↓	RR (%) ↑
GeoTrans	42	1.26	70.6
RoITr	30	1.52	20.9
SE3ET-I2 (Ours)	**17**	**0.71**	**75.9**

4.4.2 Outdoor Benchmark: KITTI Point Clouds

In addition to indoor point clouds, we evaluate our framework on outdoor LiDAR point clouds from the KITTI odometry dataset [29]. KITTI includes 11 sequences of LiDAR outdoor scans that are collected by Velodyne HDL-64E. Since this work focuses on addressing the low overlap point cloud registration challenge, we follow [63, 34, 16] to create point cloud pairs that are the first taken at least 10 meters apart within each sequence. Thus, this is not the complete KITTI odometry benchmark and the evaluation method is also different from standard KITTI odometry metrics. We use sequences 00-05 for training, 06-07 for validation, and 08-10 for testing.

For training on KITTI, we use the octahedral rotation group and set $A = 6$. There are five stages in the encoder, and the equivariant features output from the encoder have feature size $C = 2048$. In the transformer, we use the SE3ET-I design

Table 4.7: Runtime analysis of the proposed method. We tested on an Intel i9-10900K CPU and Nvidia GeForce RTX 3090 GPU.

Method	Number of Parameters			Runtime
	Backbone	Transformer	Total	(ms)
GeoTrans [63]	6.01 M	3.82 M	9.83 M	84
RoITr [89]	5.21 M	4.81 M	10.02 M	232
SE3ET-E / SE3ET-E2 (Ours)	10.10 M / 2.53 M	6.98 M / 1.75 M	17.08 M/ 4.29 M	210 / 136
SE3ET-I SE3ET-I2 (Ours)	10.10 M / 2.53 M	3.82 M / 0.96 M	13.92 M / 3.49 M	157 / 105

described in Section 4.3.2, three equivariant self-attention layers, and three invariant cross-attention layers interleaved together. The model is trained for 160 epochs with an initial learning rate of 10^{-4}.

Again, we evaluate the methods using RTE and RRE. In addition, we evaluate using *Registration Recall (RR)*: the percentage of point cloud pairs of which the RRE is smaller than 5 degrees and the RTE is smaller than 2 meters.

We compare with state-of-the-art point cloud registration methods in Table 4.5. Among other methods, our approach delivers the smallest RTE and RRE. Although we didn't achieve the top performance in the RR metric, we are only 0.1 percentage points behind, with the RR nearing its saturation point at 100%. It shows the superior performance of our method in outdoor driving scenes.

4.4.3 Generalization Experiment

In addition to the indoor and outdoor experiments, we aim to test how the proposed method generalizes to unseen data. Thus, we evaluate the pre-trained weights learned from the indoor RGB-D point cloud (3DMatch) in Section 4.4.1 on outdoor LiDAR point clouds from KITTI. To address scaling issues between the two sensor measurements, we discard points beyond the 30-meter range and scale the KITTI point clouds by a factor of 0.1 to match the 3DMatch point cloud scale.

We evaluate the performance using RTE and RRE. The *Registration Recall (RR)* is calculated as the percentage of scaled point cloud pairs of which the RRE is smaller than 5 degrees and the RTE is smaller than 0.2 meters (0.1 scale of the

original threshold for the KITTI experiment).

We present the result in Table 4.6 with the original KITTI point cloud scale to better compare with the result in Table 4.5. Our equivariant features allow better generalizability in cross-dataset evaluation since equivariant representations preserve more expressive features and thus provide better cross-domain generalizability than non-equivariant and invariant representations.

4.4.4 Run-time Analysis

In this section, We include details on network parameters and runtime for the proposed network. The result is shown in Table 4.7. Our proposed networks have faster runtime and fewer parameters than RoITr [89]. We can further increase efficiency by reducing the feature dimension by half in SE3ET-E2 and SE3ET-I2 (feature size $C = 512$) while maintaining similar performance (performance shown in Table 4.1). SE3ET-E2 and SE3ET-I2 use fewer parameters than GeoTrans [63] and still outperform it.

4.5 Conclusion

We have designed *SE3ET*, a low-overlap point cloud registration framework that leverages SE(3)-equivariant feature learning. Our approach enhances robustness to large transformations in low-overlap scenarios. We also propose an octahedral rotation group implementation and an equivariant transformer design to enable efficient training and improve performance. Experimental results on indoor and outdoor benchmarks demonstrate promising results across various metrics. Future work includes using the correlation between the anchor dimension of the equivariant features to obtain a coarse estimate of the rotation and reduce computation time for RANSAC. Moreover, combining this work within an equivariant place recognition framework [47] for mobile robots is an attractive future direction.

Chapter 5

Multi-Modal Fusion for Place Recognition: Integrating Visual Foundation Models with 3D Equivariant Features

Foundation models have demonstrated impressive zero-shot performance across various tasks. However, their application to 3D point cloud data, which provides essential geometric information for real-world deployment, is still underexplored. The lack of large-scale 3D data sets has limited the development of robust foundational 3D representations. To address this gap, we propose a multi-modal fusion approach that leverages vision foundation models and 3D equivariant models. By integrating these complementary forms of knowledge, our method captures both the detailed visual information from images and the spatial insights of 3D geometry. Experimental results on place recognition show that this fusion approach outperforms single-modality methods and is robust to rotation changes.

5.1 Introduction

The recent development of Multi-modal Large Language Models (MLLMs) has drawn researchers' attention to multi-modal learning, which combines multiple data types such as images, text, audio, and more [79]. Each modality has its unique strengths and limitations; hence, multi-modal algorithms leverage the complementary information from multiple modalities.

This paper focuses on two specific modalities: images and 3D point clouds.

Autonomous vehicle perception systems, which often use both images and point clouds, demonstrate the benefits of multi-modal approaches. While images provide rich semantic and instance-level information, point clouds offer precise 3D geometric data, enabling a more robust geometric understanding.

Visual foundation models, such as CLIP [65], DINOv2 [57], and ImageBind [30], are deep neural networks that are trained on internet-scale of image data. These models excel in generating robust visual features for varied downstream tasks such as semantic and instance segmentation, classification, object detection, and image captioning. Despite substantial advancements in 2D computer vision driven by visual foundation models, these improvements have not fully translated to 3D computer vision, primarily due to the scarcity of large-scale 3D data.

3D point clouds offer precise geometric details that remain unaffected by changes in illumination or appearance. For instance, as shown in Figure 5.1, two images and their corresponding point clouds of the same scene captured during the day and night demonstrate this phenomenon. While the images exhibit significant illumination changes, which may challenge place recognition algorithms, the point cloud measurements remain consistent.

Equivariant learning from 3D point clouds has shown promising generalization capabilities with limited data and is particularly beneficial for place recognition [46]. However, integrating images with equivariant point cloud features while preserving their equivariance is challenging due to the group structure that is added to the network architecture.

To address this challenge, we propose methods to fuse features from visual foundation models with 3D equivariant features. We extract image features from open-source pre-trained visual foundation models and 3D equivariant features from equivariant neural networks designed for 3D point clouds.

Our main contributions are as follows:

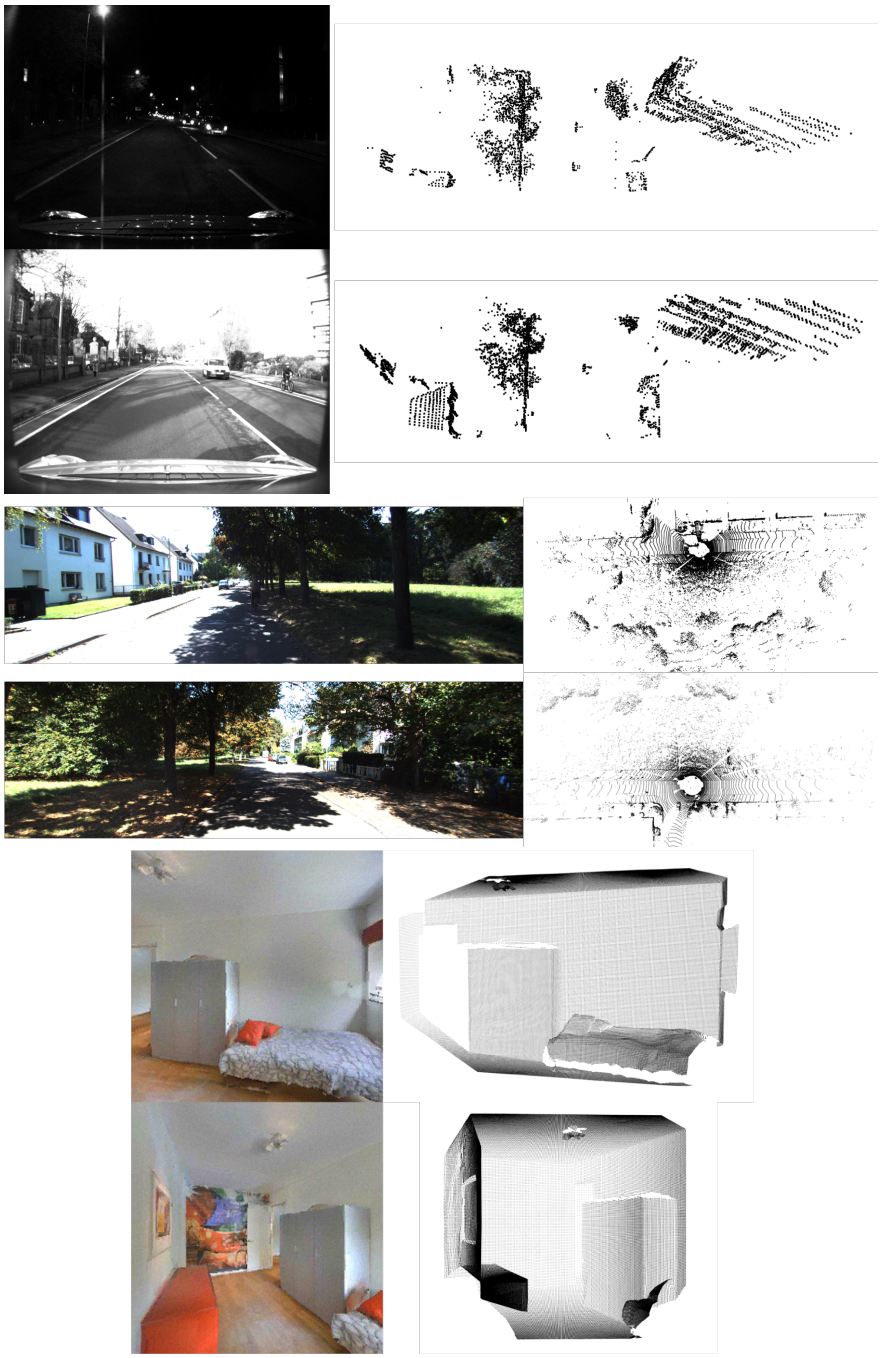


Figure 5.1: Differences between two modalities: image and point cloud. These images and point clouds are from the Oxford RobotCar [54] (top), KITTI odometry [29] (middle), and HPointLoc [91] (bottom), where each pair of input are from the same place. Point cloud data remain consistent despite illumination and viewpoint changes, whereas image data provide additional information, such as road lane lines and paintings in the room.

1. **Novel Multi-modal Fusion Framework:** To the best of our knowledge, this is the first work to introduce the fusion of image features with 3D equivariant features in a multi-modal framework.

2. **Invariant and Equivariant Fusion Techniques:** We propose invariant and equivariant fusion for stable feature representation that accounts for correspondence between 3D points and image patches. Different fusion strategies can be applied based on the need of the application.

3. **Training-free Implementation:** Our proposed methods do not require additional training, significantly simplifying the implementation process and making it accessible for broader use in practical applications.

4. **Performance Evaluation on Synthetic and Real-world Datasets:** We conduct comprehensive experiments on synthetic RGB-D indoor and real-world outdoor datasets, demonstrating our approach's effectiveness and robustness in real-world scenarios.

5. **Open-source Availability:** Open-source code is available at `https://github.com/UMich-CURLY/equivariant_foundation_model_fusion`.

5.2 Methodology

The multi-modal sensor input includes images and point clouds. First, we introduce how the features are obtained from visual foundation models and 3D equivariant networks. Then, we introduce the fusing approaches.

5.2.1 Visual Foundation Model Image Features

A visual foundation model [65, 57] learns a global descriptor $\mathbf{F}_{\mathcal{I}}^{g} \in \mathbb{R}^{D_{\mathcal{I}}}$ from the input image $\mathcal{I}$, where $D_{\mathcal{I}}$ is the global feature dimension for images. We can

obtain local features $\mathbf{F}_{\mathcal{I}}^{l} \in \mathbb{R}^{M \times C_{\mathcal{I}}}$ before global pooling, where M is the number of tokens (image patches) that feeds into the transformer-based visual encoder and $C_{\mathcal{I}}$ is the local feature dimension for images.

5.2.2 Equivariant Point Cloud Features

Group convolution is a type of convolutional operation that applies not only spatial filters but also filters along a group of transformations, here the special Euclidean group, SE(3), which includes both rotations and translations in 3D space. An equivariant network that is constructed by group convolution [101] for 3D point cloud $\mathcal{P} = \{\mathbf{p_i} \in \mathbb{R}^3 \mid i = 1, \ldots, N\}$ learns equivariant local feature $\mathbf{F}_{\mathcal{P}}^{le} \in \mathbb{R}^{N \times G \times C_{\mathcal{P}}}$ that is equivariant to SE(3) group, where N is the point cloud size, G is the group dimension, and $C_{\mathcal{P}}$ is the local feature size for point cloud. Different from non-equivariant features, the local equivariant feature $\mathbf{F}_{\mathcal{P}}^{le}$ has an extra dimension G, which corresponds to the group dimension and holds the features for various orientations.

This equivariant local feature can be converted into SE(3)-invariant by a max-pooling operation over the group dimension G. By taking the maximum over G, the resulting feature $\mathbf{F}_{\mathcal{P}}^{li} \in \mathbb{R}^{N \times C_{\mathcal{P}}}$ discards orientation-specific information and retains only the most salient aspects of each local feature. This makes $\mathbf{F}_{\mathcal{P}}^{li}$ invariant to SE(3) transformations so that the features do not change if the point cloud is rotated or translated.

Once local features are made invariant to SE(3), they can be further aggregated into a global descriptor for the entire point cloud. Generalized Mean (GeM) [64] is used to aggregate these local invariant features into a global descriptor $\mathbf{F}_{\mathcal{P}}^{gi} \in \mathbb{R}^{D_{\mathcal{P}}}$, where the $D_{\mathcal{P}}$ is the descriptor size, which represents the point cloud as a whole and is also invariant to transformations. GeM pooling allows for a flexible aggregation by adjusting the pooling parameters, enhancing the global descriptor's discriminative

power.

5.2.3 Fusing Approaches

There are several fusing approaches. One is the direct sum, in which we fuse global features from both the image and point cloud. We view this as a baseline. The second is invariant fusion, in which we fuse the invariant feature from point clouds with image features. The third one is an equivariant fusion, where we utilize the point-patch correspondence to fuse the equivariant point cloud feature with the image feature. We illustrate in Figure 5.2 and will introduce them in the following sections.

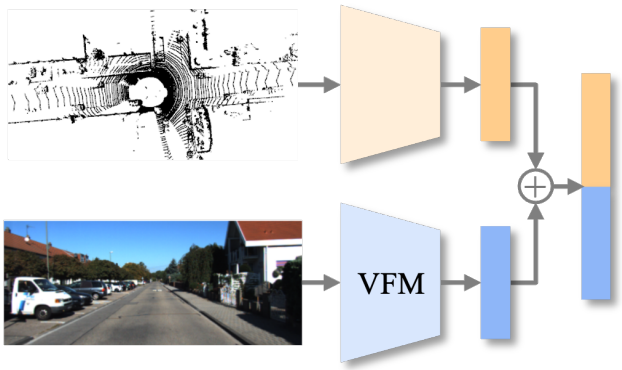
5.2.3.1 Approach 1: Direct Sum

One direct approach is to concatenate the two global features from the equivariant point cloud model and the visual foundation model together such that $\mathbf{F}_{\oplus} = \mathbf{F}_{\mathcal{P}}^{gi} \oplus \mathbf{F}_{\mathcal{I}}^{g}$. While the point cloud feature maintains SE(3)-invariance, the concatenated feature, as a whole, does not. This is because the image feature is not invariant to SE(3), and their direct sum does not inherently capture the invariance property.

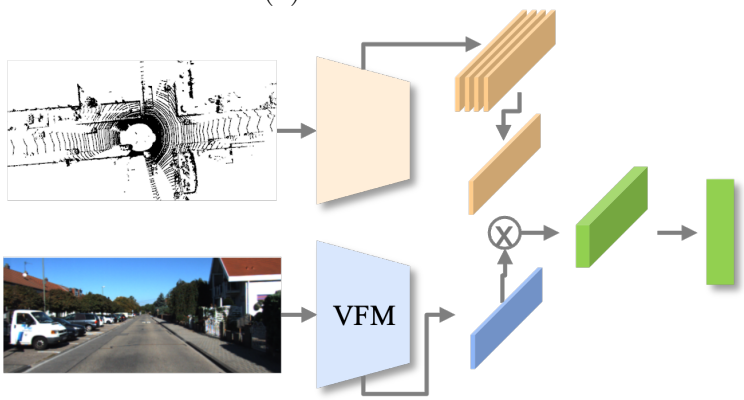
5.2.3.2 Approach 2: Invariant Fusion

The second approach is to fuse the local image feature with the local invariant point cloud feature after finding the correspondence on the local level. Let $\mathbf{A} \in \mathbb{R}^{N \times M}$ be the correspondence between m-th image patch to n-th point, such that the image feature after matching with each point $\hat{\mathbf{F}}_{\mathcal{I}}^{l} = \mathbf{A}\mathbf{F}_{\mathcal{I}}^{l} \in \mathbb{R}^{N \times C_{\mathcal{I}}}$ has the same local dimension with the point cloud feature $\mathbf{F}_{\mathcal{P}}^{li} \in \mathbb{R}^{N \times C_{\mathcal{P}}}$. We then fuse these two features with Equation (5.1),

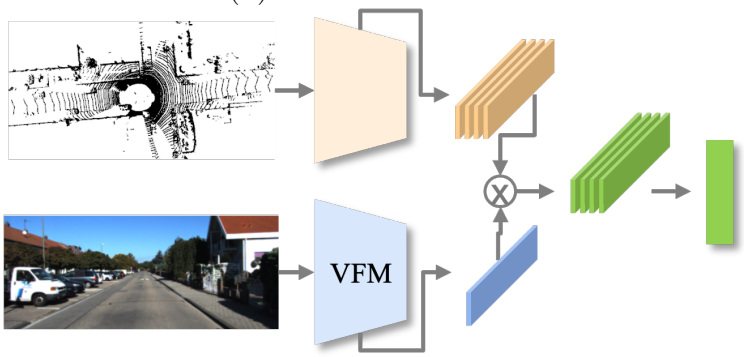
$$\mathbf{F}_{\text{inv}} = \frac{1}{N} \sum_{n} \mathbf{F}_{\mathcal{P},n}^{li} \otimes \hat{\mathbf{F}}_{\mathcal{I},n}^{l} \tag{5.1}$$



(a) Direct Sum



(b) Invariant Fusion



(c) Equivariant Fusion

Figure 5.2: Different fusion methods. Direct sum combines two modalities when they are both global representations. Invariant fusion integrates two modalities on a local scale. Equivariant fusion preserves the equivariant features from 3D data and obtains global representation at a later stage.

where $\otimes$ denotes the tensor product between point features $\mathbb{R}^{C_\mathcal{P}}$ and image features $\mathbb{R}^{C_\mathcal{I}}$.

This approach allows the invariant point cloud feature to provide geometric information to the fused feature. By combining the patch-based features with point-wise geometric anchors, this approach effectively creates a cross-referenced feature space that is more expressive. The fusion preserves geometric information from invariant point cloud features. Although this approach enhances the feature's ability to incorporate geometric information, it may not fully utilize the equivariance present in the initial equivariant feature.

5.2.3.3 Approach 3: Equivariant Fusion

The third approach expands the previous method by incorporating an additional *geometric* dimension, which is the group dimension in the equivariant point cloud feature. Using the same correspondence $\mathbf{A}$, with equivariant point cloud feature $\mathbf{F}^{le}_{\mathcal{P}}$ and local image feature $\mathbf{F}^{l}_{\mathcal{I}}$, we fuse the two features using Equation (5.2),

$$\mathbf{F}_{\text{equi},a} = \frac{1}{N} \sum_{n} \mathbf{F}^{le}_{\mathcal{P},n,a} \otimes \hat{\mathbf{F}}^{l}_{\mathcal{I},n} \tag{5.2}$$

where $\otimes$ denotes the tensor product between point features $\mathbb{R}^{C_\mathcal{P}}$ and image features $\mathbb{R}^{C_\mathcal{I}}$. a is each group element.

This approach enables equivariant fusion by introducing the symmetry structure of mathematical groups into the feature representation. In this third approach, including the group dimension G ensures that the feature representation respects the transformations under the group SE(3). In addition, the correspondence aligns local features across modalities (image and point cloud), ensuring that the equivariance from the point cloud features is preserved in the fused feature space. Therefore, through careful preservation and leveraging of the group structure during feature

fusion, equivariant fusion successfully maintains the equivariance properties essential for adapting to varying orientations and transformations.

After obtaining the fused equivariant feature, invariant pooling can be applied. This means the final feature representation $\tilde{\mathbf{F}}_{\text{equi}}$ is robust to the various poses or orientations.

$$\tilde{\mathbf{F}}_{\text{equi}} = \max_{g} \mathbf{F}_{\text{equi},g} \tag{5.3}$$

5.3 Experiments

We evaluate the performance of multi-modal fusion on the place recognition task. Place recognition, also known as loop closure detection, identifies whether the current measurement has been seen before. If the input measurement is just images, it is similar to image retrieval. To test the performance, we conducted experiments on the HPointLoc data set [91], Oxford RobotCar data set [54], and KITTI data set [29].

5.3.1 HPointLoc Data Set

The HPointLoc dataset [91] provides synthetic RGB-D images for indoor environments based on scenes from the Matterport3D dataset [10]. Key poses within each scene are spaced 2 meters apart. Six images are captured at 60-degree intervals at each key pose to create a panoramic view for the database. Up to 50 query poses are randomly generated for every key pose, with those within a 0.5-meter radius of the key pose considered to represent the same place. Each RGB and depth image pair has a 90-degree field of view. We tested on scene `1LXtFkjw3qL` with 138 data in databases and 950 data for queries.

To evaluate, we first extract features for each image in the database (six images per key pose) and store them in a KDTree. For each query input, we compute its feature and retrieve the top 16 nearest neighbors in the KDTree. If the retrieved

Data set **Method**	HPointLoc **Top-1 Recall (%)**	Oxford **Top-1% Recall (%)**	KITTI **Top-1% Recall (%)**
Point cloud only [46]	39.16	95.47	79.00
Image only	76.42	22.87	80.00
Direct sum	76.63	**95.69**	**81.00**
Invariant fusion	**77.37**	-	-
Equivariant fusion	76.84	-	-

Table 5.1: Place recognition results on three data sets.

feature is from the same key pose, we identify it as a true positive sample. The evaluation metric we use is recall, mathematically defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.4}$$

The average recall curve in Figure 5.3 illustrates performance across the top 16 matches. High recall indicates the model successfully identifies most of the relevant places. In Top-1 recall, shown in Table 5.1, the metric measures the accuracy of the model's top candidate match.

The point cloud-only method performs sub-optimally on this dataset, likely due to several factors. First, the point cloud model was trained exclusively on an outdoor dataset, making indoor point clouds an unseen domain. Indoor environments often have flat surfaces (such as walls, floors, and ceilings) and repetitive structures (like doors, windows, and furniture), offering fewer distinct depth cues. This lack of unique depth features, combined with a limited field of view and smoother depth because of the synthetic data, makes it challenging for point cloud-only methods to distinguish similar areas, leading to reduced localization accuracy. However, performance improves significantly when the point cloud features are fused with image features, as the visual cues from images help overcome these limitations.

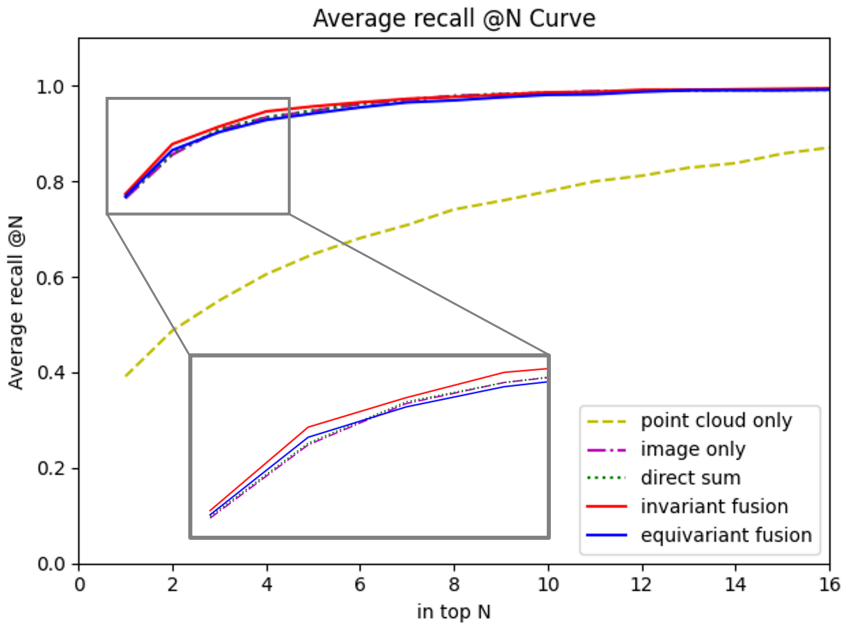


Figure 5.3: Place recognition performance on HPointLoc dataset.

5.3.2 Oxford RobotCar Data Set

The Oxford RobotCar data set [54] 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. We test on 3 sequences: one sequence at night and two during daytime, as shown in Figure 5.1, making the image model more challenging.

We think the poor performance of the image model is because of the significant illumination change in the testing sequences, in which one is daytime and one is nighttime, and the appearance is drastically different. This result shows that the fusion method performs better than the single-modal methods.

5.3.3 KITTI Odeomtry Data Set

We also evaluate KITTI [29] sequence 08 with opposite orientation loop closure to test how well the model generalizes. Figure 5.5 shows the average recall curve.

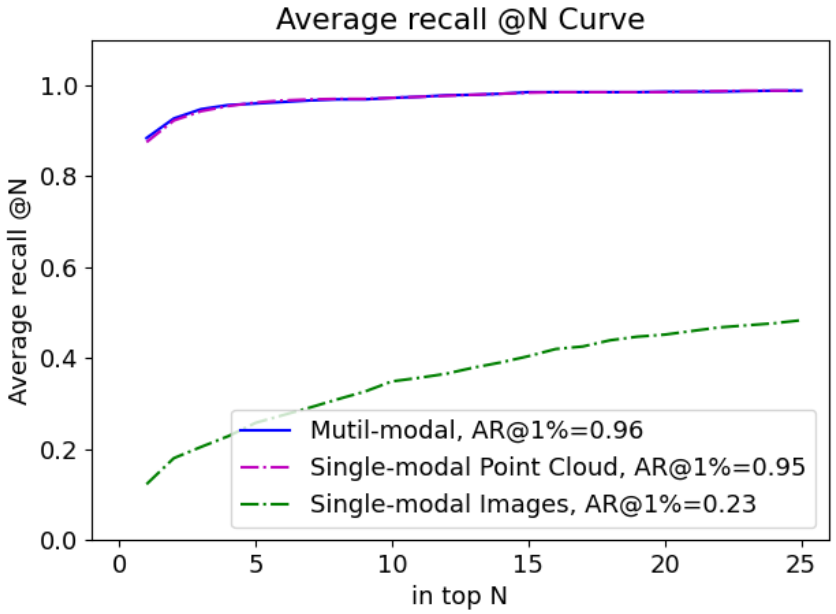


Figure 5.4: Average recall curve for performing place recognition on Oxford RobotCar dataset sequence 5, 6, and 7 using the direct sum approach. The multi-modal result is better than any single-modal.

The point cloud model is not trained on this dataset but is still generalizable and performs well on challenging reverse loop closure because of its SE(3)-invariant property. The image model performs better on KITTI than Oxford because KITTI is a consecutive sequence. This experiment shows that multi-modal performance is slightly better than any single-modal method.

5.3.4 Invariance Experiment

We further investigate the robustness of the fused features by conducting an invariance experiment. In this experiment, we rotate images and point clouds by 90°, 180°, and 270° along the z-axis (as shown in Fig. 5.6). We then extract features from the rotated inputs and apply the proposed fusion methods. A robust fusion method should produce features from the rotated inputs that closely resemble those obtained from the original (0° rotated) inputs. As shown in Fig. 5.7, the point cloud-only method, leveraging an SE(3)-invariant framework, consistently maintains the same

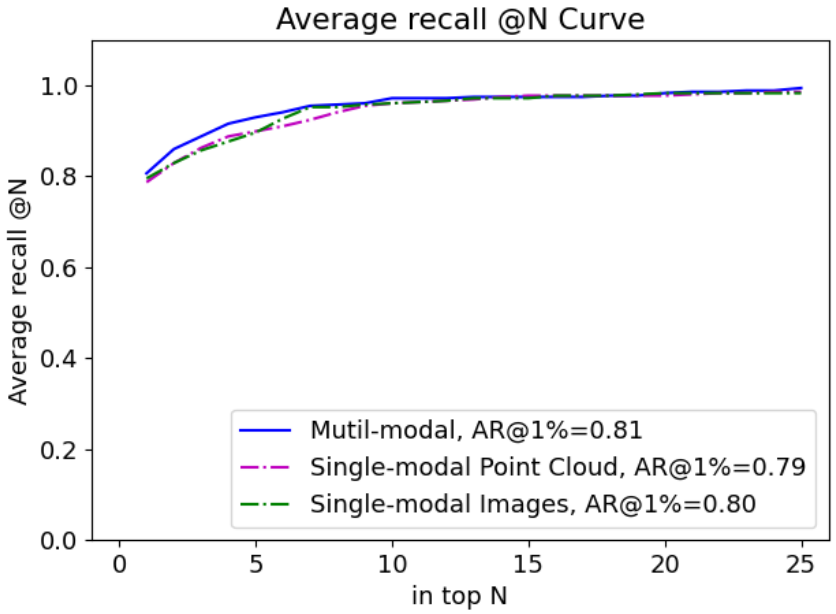


Figure 5.5: Average recall curve for performing place recognition on KITTI sequence 08 using the direct sum approach. The multi-modal result is better than any single-modal method.

representation regardless of rotation. In contrast, the image-only method produces inconsistent representations. Among the proposed multi-modal fusion approaches, equivariant fusion demonstrates superior performance compared to invariant fusion and direct summation. These results highlight that the proposed fusion method enhances the robustness of feature representations under rotational transformations.

5.4 Limitation

There might be viewpoint-changing challenges for the image-based approach or when using RGB-D point clouds. The change in the view when the observer (robot) rotates is drastic when the field of view is limited. The invariance of the features could be improved if the input images are multi-view images or panorama images by combining multiple cameras that compensate for viewpoint changes. However, this is not implemented in this work.

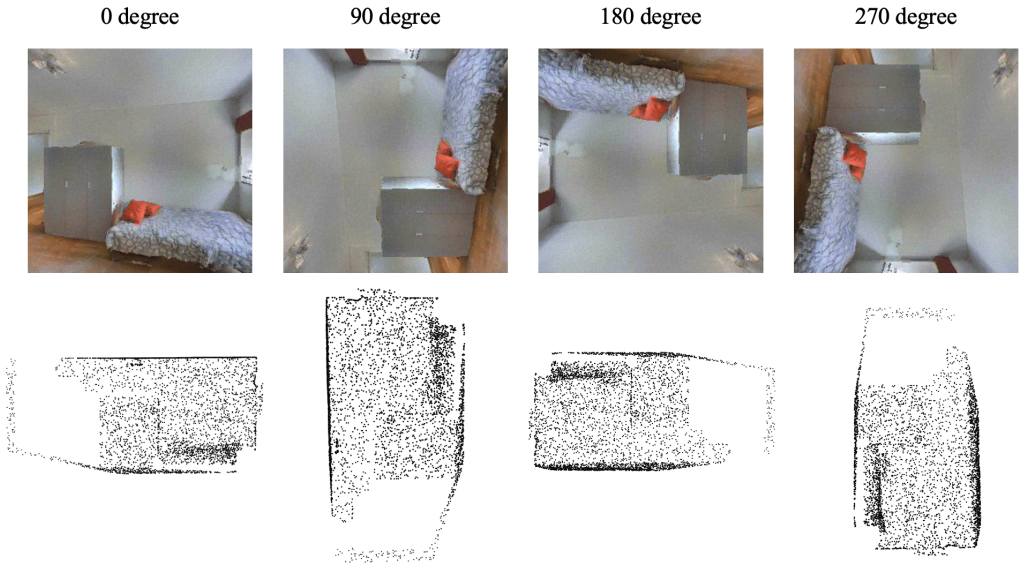


Figure 5.6: Images and point clouds from the HPointLoc data set after rotation for 90, 180, and 270 degrees.

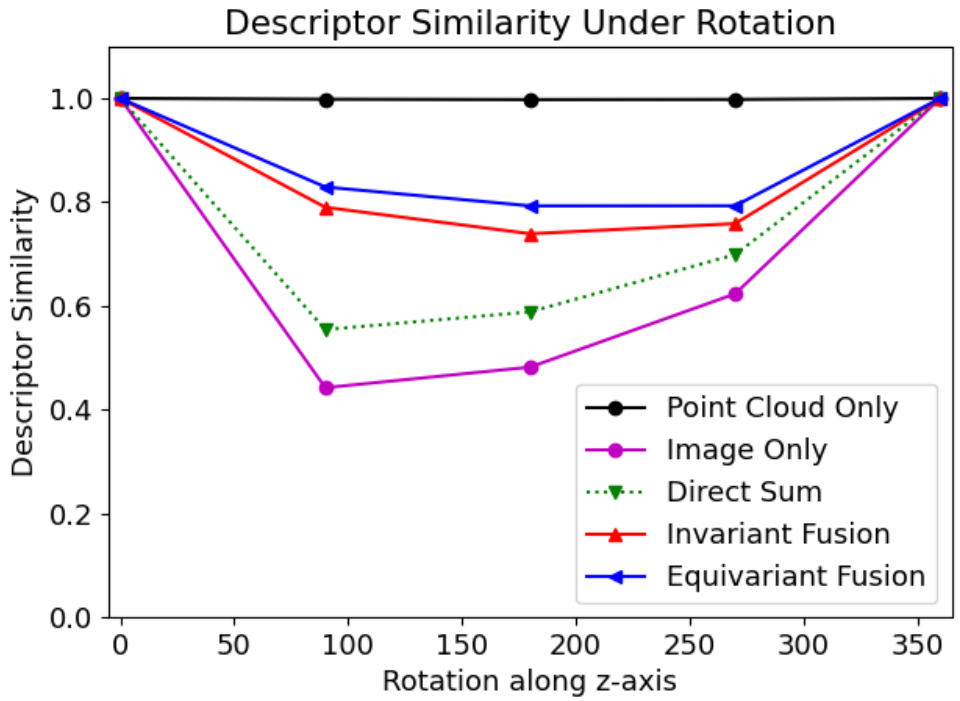


Figure 5.7: Results of the invariance experiment. Equivariant fusion is performing better than other methods.

5.5 Conclusion

This chapter introduces three ways to fuse equivariant features from point clouds with visual foundation model features. Our approach considers the group dimension in the equivariant feature to provide ways of equivariant fusion and invariant fusion. Three experiments on different data sets show that this approach can provide stable representations when two modalities have different performances on different data sets. Fusing with modalities other than images can be explored for future work.

Chapter 6

Conclusion and Future Works

In this thesis, we address the research gap in leveraging geometric learning for robot localization tasks. We present a place recognition framework that employs SE(3)-equivariant representation learning. By integrating geometric learning into the framework, we achieve a stable representation that is SE(3)-invariant, enhancing robustness to challenging scenes involving arbitrary transformations and enabling generalizability across different data sets. Additionally, we propose a low-overlap point cloud registration framework that utilizes SE(3)-equivariant feature learning. We introduce transformer mechanism designs that maintain equivariant features and learn correspondences across discretized rotation groups, thus improving robustness to significant transformations in low-overlap scenarios. Furthermore, we develop methods to fuse equivariant features with image features. The fusion of these two modalities results in a more stable representation compared to using a single modality, as each modality offers its own distinct advantages.

However, there are some limitations to the proposed works. One limitation is the relatively slow run time and the need for optimized libraries to perform real-time robot localization. However, with the advancement of more powerful computing hardware in the near future, we expect this limitation to be largely resolved in the near future. In addition, this thesis does not explore the potential of equivariant encoders under other Lie groups, such as those enabling invariance to scale and deformation. This represents an interesting direction for future research that could

further enhance the robustness and applicability of the proposed frameworks. Another limitation is the performance of the algorithm under conditions of occlusion, which is not explored in this thesis. Occlusion can significantly affect the accuracy and reliability of place recognition and point cloud registration, as partially visible objects or environments may lead to incorrect localization or mapping. Investigating how the proposed frameworks handle occluded scenes and developing strategies to mitigate the impact of occlusion would be an important area for future research, to enhance the robustness of the approaches in real-world scenarios.

In the future, combining the proposed works on place recognition and point cloud registration for mobile robots presents an attractive direction. Leveraging the correlation between the group dimension of the equivariant features to obtain a coarse estimate of the rotation could reduce computation time for RANSAC, offering another promising avenue for future exploration. Furthermore, incorporating various modalities such as audio, thermal, and force sensors would be a compelling direction, given that robots are often equipped with a diverse array of sensors. This multi-modal fusion could enhance the robustness and versatility of the localization systems, enabling them to perform more reliably in complex and dynamic environments. Extending the work to include scene understanding and object representation for robot manipulation is also a promising direction. By integrating the proposed SE(3)-equivariant representation learning with techniques for scene understanding, robots could achieve a more comprehensive perception of their environment, enabling them to recognize and interact with objects more effectively. This could benefit tasks such as object detection, classification, and pose estimation, ultimately enhancing the robot's ability to perform complex manipulation tasks. Combining these domains could lead to more autonomous and capable robotic systems capable of a wide range of applications.

BIBLIOGRAPHY

[1] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1255–1264, 2023.

[2] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[5] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[6] PJ Besl and ND McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(02):239–256, 1992.

[7] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar-camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[8] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 13229–13238, 2021.

[9] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. LCDNet: Deep loop closure detection for LiDAR SLAM based on unbalanced optimal transport. *arXiv preprint arXiv:2103.05056*, 2021.

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[11] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[12] Evangelos Chatzipantazis, Stefanos Pertigkiozoglou, Edgar Dobriban, and Kostas Daniilidis. SE(3)-equivariant attention networks for shape reconstruction in function space. In *International Conference on Learning Representations*, 2023.

[13] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall 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.

[14] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall 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.

[15] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, Cyrill Stachniss, and FKIE Fraunhofer. OverlapNet: Loop closing for LiDAR-based SLAM. In *Proceedings of the Robotics: Science and Systems Conference*, 2020.

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

[17] William Clark, Maani Ghaffari, and Anthony Bloch. Nonparametric continuous sensor registration. *Journal of Machine Learning Research*, 22(271):1–50, 2021.

[18] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2021.

[19] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

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

[21] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. In *International Conference on Machine Learning*, pages 7694–7731. PMLR, 2023.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

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

[24] Yongzhi Fan, Xin Du, and Jizhong Shen. Fresco: Frequency-domain scan context for lidar-based place recognition with translation and rotation invariance. *arXiv preprint arXiv:2206.12628*, 2022.

[25] Zhaoxin Fan, Zhenbo Song, Wenping Zhang, Hongyan Liu, Jun He, and Xiaoyong Du. Rpr-net: A point cloud-based rotation-aware large scale place recognition network. *arXiv preprint arXiv:2108.12790*, 2022.

[26] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[27] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8893–8902, 2021.

[28] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3d roto-translation equivariant attention networks. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 33:1970–1981, 2020.

[29] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[30] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[31] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. *arXiv preprint arXiv:2207.11514*, 2022.

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

[33] Li He, Xiaolong Wang, and Hong Zhang. M2dp: A novel 3D point cloud descriptor and its application in loop closure detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 231–237. IEEE, 2016.

[34] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021.

[35] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023.

[36] Xiaoshui Huang, Guofeng Mei, Jian Zhang, and Rana Abbas. A comprehensive survey on point cloud registration. *arXiv preprint arXiv:2103.02690*, 2021.

[37] Xiaoshui Huang, Yangfu Wang, Sheng Li, Guofeng Mei, Zongyi Xu, Yucheng Wang, Jian Zhang, and Mohammed Bennamoun. Robust real-world point cloud registration by inlier detection. *Computer Vision and Image Understanding*, 224:103556, 2022.

[38] Hordur Johannsson, Michael Kaess, Maurice Fallon, and John J Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 54–61. IEEE, 2013.

[39] Giseop Kim, Sunwook Choi, and Ayoung Kim. Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 2021.

[40] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4802–4809. IEEE, 2018.

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

[42] Jacek 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.

[43] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[44] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. RINet: Efficient 3D lidar-based place recognition using rotation invariant neural network. *IEEE Robotics and Automation Letters*, 7(2):4321–4328, 2022.

[45] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022.

[46] Chien Erh Lin, Jingwei Song, Ray Zhang, Minghan Zhu, and Maani Ghaffari. SE(3)-equivariant point cloud-based place recognition. In *Conference on Robot Learning*, pages 1520–1530. PMLR, 2023.

[47] Chien Erh Lin, Jingwei Song, Ray Zhang, Minghan Zhu, and Maani Ghaffari. SE(3)-equivariant point cloud-based place recognition. In *Proc. 6th Conf. Robot Learn.*, pages 1520–1530. PMLR, 2023.

[48] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[49] Tzu-Yuan Lin, Minghan Zhu, and Maani Ghaffari. Lie Neurons: Adjoint-equivariant neural networks for semisimple Lie algebras. In *Forty-first International Conference on Machine Learning*, 2024.

[50] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3D point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2831–2840, 2019.

[51] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[52] Sha Lu, Xuecheng Xu, Huan Yin, Rong Xiong, and Yue Wang. One ring to rule them all: Radon sinogram for place recognition, orientation and translation estimation. *arXiv preprint arXiv:2204.07992*, 2022.

[53] Junyi Ma, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, and Xieyuanli Chen. OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition. *IEEE Robotics and Automation Letters*, 2022.

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

[55] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021.

[56] Guofeng Mei, Hao Tang, Xiaoshui Huang, Weijie Wang, Juan Liu, Jian Zhang, Luc Van Gool, and Qiang Wu. Unsupervised deep probabilistic approach for partial point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13611–13620, 2023.

[57] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[58] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393. IEEE, 2020.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith 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.

[60] 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.

[61] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

[62] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[63] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022.

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

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

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

[67] Martial Sanfourche, Vincent Vittori, and Guy 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.

[68] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

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

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 30, 2017.

[71] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0:

Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[72] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1630–1641, 2022.

[73] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[74] Ying Wang, Zezhou Sun, Cheng-Zhong Xu, Sanjay Sarma, Jian Yang, and Hui Kong. Lidar iris for loop-closure detection. *arXiv preprint arXiv:1912.03825*, 2019.

[75] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3523–3532, 2019.

[76] Yujie Wang, Chenggang Yan, Yutong Feng, Shaoyi Du, Qionghai Dai, and Yue Gao. Storm: Structure-based overlap matching for partial point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[77] Zhongli Wang, Yan Shen, Baigen Cai, and Muhammad Tariq Saleem. A brief review on loop closure detection with 3d point cloud. In *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 929–934. IEEE, 2019.

[78] Louis Wiesmann, Rodrigo Marcuzzi, Cyrill Stachniss, and Jens Behley. Retriever: Point cloud retrieval in compressed 3D maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2022.

[79] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and S Yu Philip. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023.

[80] Florian Wulff, Bernd Schäufele, Oliver Sawade, Daniel Becker, Birgit Henke, and Ilja Radusch. Early fusion of camera and lidar for robust road detection based on u-net fcn. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1426–1431. IEEE, 2018.

[81] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11348–11357, 2021.

[82] Dong Xu, Jingbin Liu, Juha Hyyppä, Yifan Liang, and Wuyong 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.

[83] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.

[84] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11824–11833, 2020.

[85] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022.

[86] Huan Yin, Yue Wang, Xiaqing Ding, Li Tang, Shoudong Huang, and Rong Xiong. 3D lidar-based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1380–1392, 2019.

[87] Peng Yin, Fuying Wang, Anton Egorov, Jiafan Hou, Ji Zhang, and Howie 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.

[88] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 34:23872–23884, 2021.

[89] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5384–5393, 2023.

[90] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17702–17711, 2023.

[91] Dmitry Yudin, Yaroslav Solomentsev, Ruslan Musaev, Aleksei Staroverov, and Aleksandr I Panov. Hpointloc: point-based indoor place recognition using synthetic rgb-d images. In *International Conference on Neural Information Processing*, pages 471–484. Springer, 2022.

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

[93] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15244–15253, 2023.

[94] Ray Zhang, Tzu-Yuan Lin, Chien Erh Lin, Steven A Parkison, William Clark, Jessy W Grizzle, Ryan M Eustice, and Maani Ghaffari. A new framework for registration of semantic point clouds from stereo and RGB-D cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 12214–12221. IEEE, 2021.

[95] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022.

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

[97] Xiwu Zhang, Lei Wang, and Yan Su. Visual place recognition: A survey from deep learning perspective. *Pattern Recognition*, 113:107760, 2021.

[98] Hengwang Zhao, Hanyang Zhuang, Chunxiang Wang, and Ming Yang. G3doa: Generalizable 3d descriptor with overlap attention for point cloud registration. *IEEE Robotics and Automation Letters*, 7(2):2541–2548, 2022.

[99] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

[100] Lifa Zhu, Dongrui Liu, Changwei Lin, Rui Yan, Francisco Gómez-Fernández, Ninghua Yang, and Ziyong Feng. Point cloud registration using representative overlapping points. *arXiv preprint arXiv:2107.02583*, 2021.

[101] Minghan Zhu, Maani Ghaffari, William A Clark, and Huei Peng. E2PN: Efficient SE(3)-equivariant point network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.