2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Stabilize an Unsupervised Feature Learning for LiDAR-based Place Recognition

Peng Yin, Lingyun Xu, Zhe Liu, Lu Li, Hadi Salman, Yuqing He,
Weiliang Xu, *Senior Member, IEEE,* Hesheng Wang, *Senior Member, IEEE,* and Howie Choset, *Fellow, IEEE*

***Abstract*— Place recognition is one of the major challenges for the LiDAR-based effective localization and mapping task. Traditional methods are usually relying on geometry matching to achieve place recognition, where a global geometry map need to be restored. In this paper, we accomplish the place recognition task based on an end-to-end feature learning framework with the LiDAR inputs. This method consists of two core modules, a dynamic octree mapping module that generates local 2D maps with the consideration of the robot's motion; and an unsupervised place feature learning module which is an improved adversarial feature learning network with additional assistance for the long-term place recognition requirement. More specially, in place feature learning, we present an additional Generative Adversarial Network with a designed Conditional Entropy Reduction module to stabilize the feature learning process in an unsupervised manner. We evaluate the proposed method on the *Kitti* dataset and *North Campus Long-Term* LiDAR dataset. Experimental results show that the proposed method outperforms state-of-the-art in place recognition tasks under long-term applications. What's more, the feature size and inference efficiency in the proposed method are applicable in real-time performance on practical robotic platforms.**

I. INTRODUCTION

In the last decade, the robotic community has had a lot of breakthroughs and developments. One of the major domains that improved substantially is simultaneous localization and mapping (SLAM), which made real-world applications such as autonomous driving possible. Effective localization and mapping highly rely on robust place recognitions (PR) or loop closure detection (LCD) abilities, especially for long-term navigation tasks [1], [2]. This is one of the major challenges for current SLAM systems. Visual place recognition is a PR method that uses a camera for the task of matching two scenes. The problem of visual PR is challenging due to the fact that same scene appears differently under different season or weather conditions. Besides, the same scene place appears different from different viewpoints, which occurs very often during SLAM process because there is no guarantee that a robot will observe each local scene always from the same viewpoint. These are all challenges faced by a robot performing SLAM.

LiDAR-based place recognition is another way of performing the PR where only LiDAR measurements are used to extract features of the place. Compared with visual data, LiDAR data is more robust to lighting and appearance changes. However extracting efficient place descriptor from raw 3D point-clouds in real time is intractable. Also storing all the 3D points into a joint point-cloud overloads storage and is computationally inefficient for long-term navigation tasks. LiDAR odometry [3] could achieve accurate long-term odometry results on the well-known *Kitti* [4] benchmark, yet there is no efficient place recognition method for LiDAR-based SLAM framework. Therefore robust and efficient place recognition using LiDAR data is an interesting problem which has become the bottleneck for LiDAR-based SLAM approaches.

Recently, Google's cartographer used the 2D bird-view map of LiDAR inputs and the depth-first search to achieve large scale place recognition [5]. This method is sensitive to viewpoint (heading/orientation) changes. Another method, NOctoSLAM [6] introduced a fast octree structure to solve the place recognition tasks using LiDAR inputs. The complexity of loop closure detection tends to grow with the map scale, and matching all the current scan points with potential target point-clouds is computationally expensive. Thus, instead of extracting the whole points in a current scan for geometry matching, SegMatch [7] proposed a matching-by-segmentation method where static objects are segmented out as landmarks to reduce the matching complexity. The major limitation for SegMatch is the additional computation requirements for 3D point segmentation, in addition to the assumption that there is enough static objects, however, this basic assumption can not always be satisfied in the real situations.

In this paper, instead of relying on geometry based matching, we propose a LiDAR-based feature learning framework for long-term place recognition. At the core of our framework lies the idea of compressing raw LiDAR data into low dimension features for fast storage and retrieval purposes. In traditional visual-based place recognition approaches, handcrafted features [9], [8] or bag-of-words [10], to create

P. Yin, L. Xu and Y. He are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, also with the University of Chinese Academy of Sciences, Beijing, China. (pyin2@andrew.cmu.edu; 121067240@qq.com; heyuqing@sia.cn.)

Z. Liu is with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. (zheliu@cuhk.edu.hk.)

L. Li, Hadi Salman and Howie Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, USA. (luli2@cmu.edu; hadis@cmu.edu; choset@cmu.edu.)

W. Xu is with the Department of Mechanical Engineering, University of Auckland, New Zealand. (p.xu@auckland.ac.nz)

H. Wang is with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. (wanghesheng@sjtu.edu.cn)

Corresponding author: L. Xu.

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

low dimensional visual descriptors, have been successfully applied in FABMAP [12] and SeqSLAM [13]. However, raw LiDAR inputs do not have detailed texture information as visual images, and similar corner or edge features may easily lead to wrong matchings in the LiDAR-based PR.

To address this problem, we introduce a Dynamic Octree-based Mapping (DOM) module, where the local occupancy map is updated based on raw LiDAR inputs and the motion error model of the robot. Based on the octree map, we obtain a bird-view map by projecting the DOM map onto the surface plane as shown in Fig. 2. For long-term place recognition, some places may be encountered once or multiple times, thus, this requires the feature inference module to have higher generalization ability and capture the unique mapping between the original data space and the compressed latent code space. In order to achieve the long-term feature learning for PR, we utilize an improved adversarial feature learning method [15], where an additional generative adversarial networks (GAN) [16] is introduced to improve the feature inference generalization ability, and a lower bound is applied to enforce the unique mapping between the original data space and the latent code space.

The main contributions of this work can be summarized as the following three aspects: Firstly, we introduce a dynamic octree mapping method to model the local 3D map around the mobile robot with raw LiDAR inputs. The local occupancy map is updated based on an error model of the robot's motion. Secondly, we introduce a novel place feature inference method which encodes the bird-view images into a low dimensional feature vector that is fast to store and retrieve for long-term place recognition applications. This is done through: 1) utilizing a GAN module to improve the generalization ability of the feature inference module, 2) proposing a tractable method to ensures the unique mapping from the data space to the latent code space. Thirdly, we test our framework on two challenging datasets: the *Kitti* [4] dataset and the *North Campus Long-Term (NCTL)* LiDAR [2] dataset. In both datasets, our proposed method shows robustness to viewpoints changes compared to other state-of-the-art appearance based place recognition methods. Furthermore the feature inference for each frame can reach 10 Hz on a Nvidia embedded Jetson TX2 board, with a *1KB* feature code for each place. This makes the proposed method suitable for real-time applications where the computation power and storage space are key concerns.[1]

The rest of this paper is organized as follows: Section II presents the details of our proposed method. Section III shows experimental results and comparisons with state-of-the-art methods in place recognition researches on challenging datasets. Finally, Section IV concludes the proposed method and introduces the future works.

II. Proposed Method

In this section, we introduce the details of the proposed method, including the dynamic octree mapping module and the stable place feature learning module.

[1]Upon publication, the code and the datasets used in the paper are going to be open sourced.

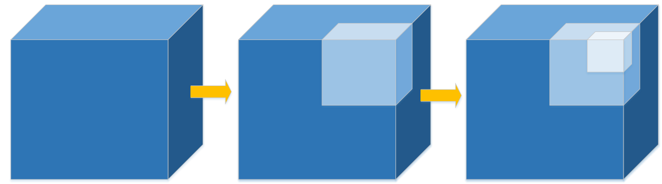


Fig. 1: Octree Structure. Each node is divided into eight sub-children with the equal sub-space.

A. Dynamic Octree Mapping (DOM)

In order to model the local mapping around a robot, we apply a dynamic octree structure which is tied to the mobile robot's position. Octree is a tree type structure: from the root node, each sub-children is assigned with equal sub-space, as shown in Fig. 1. The sub-nodes continue to be divided until the node edge of a leaf node reaches a given threshold. Instead of using all the point-clouds to build a static mapping [21], [6], the root node of our dynamic octree is fixed to the current mobile robot's position, and the entire leaf nodes is updated dynamically. DOM contains two major steps: *Map Building* and *Motion Updating*.

1) Map Building: For the lowest level leaf nodes in dynamic octree, their occupancy beliefs can be updated by the log-odds method described in [22]. For a leaf node n with a given sequence measurement $z_{1:t}$, the occupancy estimation $P(n|z_{1:t})$ can be calculated as,

$$P(n|z_{1:t}) = \\ [1 + \frac{1-P(n|z_t)}{P(n|z_t)} \cdot \frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \cdot \frac{P(n)}{1-P(n)}]^{-1} \quad (1)$$

where $P(n)$ is the initial occupancy estimation for leaf node n, and $P(n|z_t)$ is the occupancy belief based on the current observation. To simplify this equation, we can assign $P(n)$ to 0.5, then the above equation can rewritten as,

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \quad (2)$$

$$L(x) = \log[\frac{P(x)}{1-P(x)}] \quad (3)$$

where Eq.[3] shows the transformation between the log-odds and the probability. The original occupancy update problem is now transformed into a linear accumulation operation.

2) Motion Updating: Updating the position of the mobile robots introduces additional uncertainty for the occupancy of each leaf node. The motion update step should reduce occupancy beliefs for the occupied nodes and increase the occupancy for unoccupied nodes. But to determine whether a leaf node belongs to an occupied or unoccupied nodes is not an obvious task. Thus, we use the log-odds updating mechanism to settle down this problem. With a motion updating estimation $\mathcal{N}(u_t, \sigma)$, we update the log-odds of each leaf node as follows,

$$\hat{L}(n|z_{1:t}) = L(n|z_{1:t}) \cdot \mathcal{N}(u_t, \sigma) \quad (4)$$

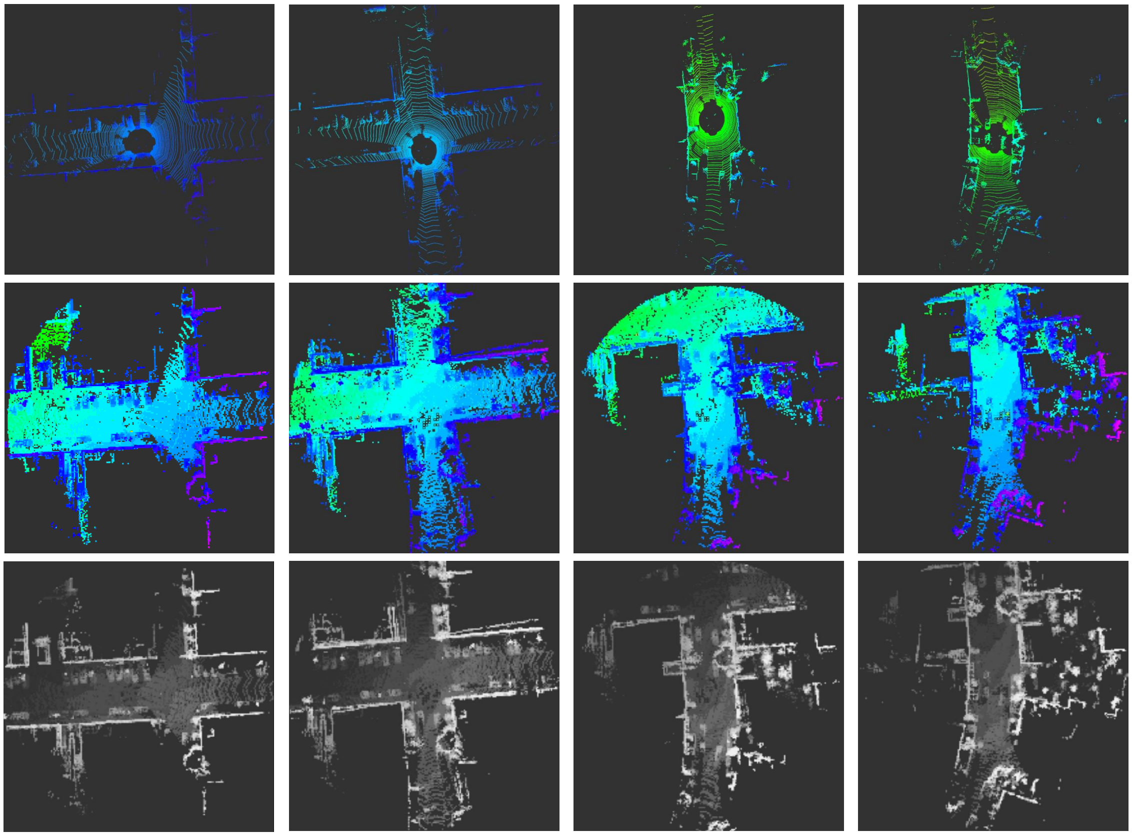


Fig. 2: Examples of the dynamic octree mapping result. The first row shows the raw point-cloud data; the second row shows the accumulated occupancy map based on the proposed dynamic octree mapping; the third row shows the projected bird-view 2D maps.

where u_t is the current position transformation and σ is the relative covariance estimated by the odometry. Based on the new log-odds $\hat{L}(n|z_{1:t})$, the occupancy $P(n|z_{1:t})$ can be updated as follows,

$$\hat{P}(n|z_{1:t}) = \frac{\exp(\hat{L}(n|z_{1:t}))}{\exp(\hat{L}(n|z_{1:t})) + 1} \tag{5}$$

since $\mathcal{N}(u_t, \sigma) \in (0,1)$, and $\|\hat{L}(n|z_{1:t})\| \leq \|L(n|z_{1:t})\|$, thus compared to the original $P(n|z_{1:t})$, we have,

$$\begin{aligned} P(n|z_{1:t}) > 0.5 \Rightarrow P(n|z_{1:t}) > \hat{P}(n|z_{1:t}) > 0.5 \\ P(n|z_{1:t}) < 0.5 \Rightarrow P(n|z_{1:t}) < \hat{P}(n|z_{1:t}) < 0.5 \end{aligned} \tag{6}$$

therefore the new updated probability $\hat{P}(n|z_{1:t})$ reduces the occupancy beliefs for occupied leaf nodes and increases the occupancy beliefs for unoccupied leaf nodes.

Fig. 2 shows the dynamic octree mapping result on the *Kitti* odometry dataset. The first row shows the raw point-cloud data from the *kitti* dateset; The second row shows the accumulated occupancy map based on the proposed dynamic octree mapping, where the root node is fixed to the robot position. The third row shows the bird-view projections of the corresponding octree maps, where only leaf nodes with occupancy beliefs greater than a fixed threshold are projected on the bird-view map.

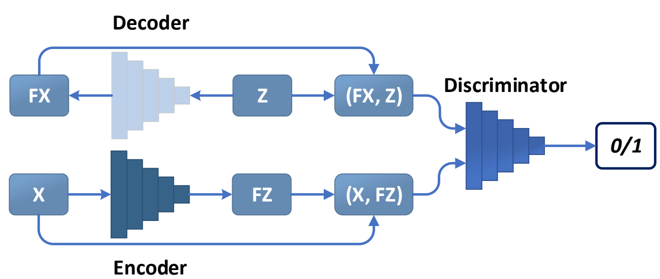


Fig. 3: Bidirectional generative adversarial networks.

B. Stable Place Feature Learning (PFL)

In order to compress the high dimensional space of the LiDAR measurements, we use an adversarial feature learning method which is a variant of the GAN approach. As shown in Fig. 3, the architecture of this method consists of a decoder model De to generate synthesis data $x \in \mathcal{X}$ from latent code $z \in \mathcal{Z}$, an encoder module En that maps the data space x to the latent code space, and a discriminator model that distinguish between the joint distribution$P(x, En(x))$ and $P(De(z), z)$. In an ideal world, the inverse mapping $x \sim De(En(x))$ could be achieved when the two joint distribution $P(x, En(x))$ and $P(De(z), z)$ are equal [15], however in practice, it is intractable to reduce the distance between these two distributions.

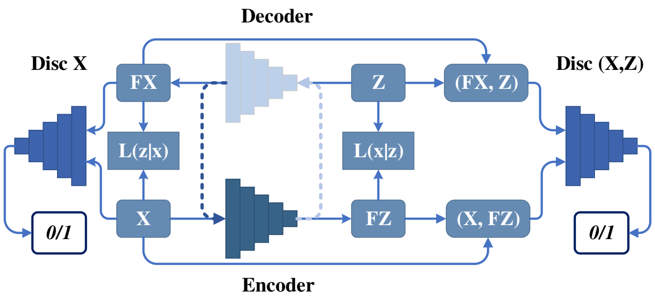


Fig. 4: The framework of Stable Adversarial Feature Learning.

In addition, for the long-term place recognition task, the encoder module should have the a high generalization and unique mapping ability. It is difficult to directly improve the generalization ability of the encoder, however we can improve it indirectly. Given the inverse mapping $x \sim De(En(x))$, a more generalized decoder De improves the generalization ability of the encoder En. Furthermore, for the real data distribution $P(x)$ and the corresponding latent codes distribution $P(z)$, the conditional entropy $H(z|x)$ measures the uncertainty of z given a data point x. The formulation $H(z|x) = H(x|z) = 0$ can be established if and only if z is the identity mapping of x. Thus, our proposed PFL module tries to improve adversarial feature learning in two aspects: improves the generalization ability in De and decreases the conditional entropy $H(z|x)$ and $H(x|z)$.

1) Improve Decoder's Generalization Ability: To improve the generalization ability of the decoder, we apply an additional GAN to enforce the learned dencoder module to capture more geometry detail. This can be achieved by defining the loss function,

$$\mathcal{L}_{GAN} = \min_{\theta,\phi} \max_{\omega} E(\log(D_\omega(x)) + E_{x\sim q_\phi(x|\tilde{z}),\tilde{z}\sim p_\theta}(\log(1 - D_\omega(x))), \tag{7}$$

where p_θ is the encoder module that maps x to z, and q_ϕ is the decoder module that generates synthetic data to fool the discriminator D_ω. θ, ϕ and ω are the network parameters correspondingly. The network module is shown in Fig. 4. As shown by [16], the optimal decoder and discriminator shorten the distance between the distribution of real data and synthesis data, and enforce the decoder module to have higher generalization ability.

2) Conditional Entropy Reduction: For conditional entropy $H(X|Z)$, we can have the following transformation,

$$\begin{aligned} H(Z|X) &= \sum_{x\in\mathcal{X}} p(x) H(Z|X = x) \\ &= -\sum_{x\in\mathcal{X}} p(x) \sum_{z\in\mathcal{Z}} p(z|x) \log p(z|x) \\ &= -\sum_{x\in\mathcal{X}, z\in\mathcal{Z}} p(x,z) \log p(z|x) \\ &= -E_{(x,z)}[\log(q_\phi(z = \hat{z}|x))] \end{aligned} \tag{8}$$

Both $H(Z|X)$ and $H(X|Z)$ are hard to evaluate directly because we cannot access the paired data (x, z). One alternative approach is to optimize the conditional entropies by minimizing their upper bounds,

$$\begin{aligned} &\inf_{\theta,\phi}\{H(z|x)\} = \inf_{\theta,\phi}\{-E_{q_\phi(x,z)}[\log(q_\phi(z|x))]\} \\ &= \inf_{\theta,\phi}\{-E_{q_\phi(x,z)}[\log(p_\theta(x|z))] \\ &- E_{q_\phi(x,z)}[\mathbf{KL}(q_\phi(x|z)\|(p_\theta(x|z)))]\} \end{aligned} \tag{9}$$

$$\begin{aligned} &\inf_{\theta,\phi}\{H(z|x)\} \leq \inf_{\theta,\phi}\{-E_{q_\phi(x,z)}[\log(p_\theta(x|z))]\} \\ &\triangleq \inf_{\theta,\phi} \mathcal{L}_{(z|x)}(\tilde{z}, \hat{z}), \end{aligned} \tag{10}$$

where $\mathbf{KL}$ is the Kullback-Leibler divergence, and $\log(q_\phi(z|x))$ is the log-likelihood of feature z under $q_\phi(z|x)$. However, we need to provide the pairs $(z, \tilde{z})$ for the log-likelihood estimation. Without loss of generality, we use the reconstruction loss $\mathcal{L}_{(z|x)}(\tilde{z}, \hat{z})$ as the upper bound of $H(z|x)$. Similarly, the upper bound of conditional entropy $H(x|z)$ could be derived as $\mathcal{L}_{(x|z)}(x, \hat{x}) = -E_{p_\theta(x,z)}[\log(q_\phi(z|x))]$.

Both $\mathcal{L}_{(x|z)}$ and $\mathcal{L}_{(z|x)}$ are shown in Fig. 4. Finally, the joint loss function combines the previously defined loss functions and is defined as,

$$\mathcal{L}_{Joint} = \mathcal{L}_{AFI} + \mathcal{L}_{GAN} + \mathcal{L}_{(x|z)} + \mathcal{L}_{(z|x)}. \tag{11}$$

where $\mathcal{L}_{AFI}$ is the original adversarial feature inference loss function.

Finally, in order to increase the robustness of the proposed method, we combine the joint heading-invariant features to further improve the performance of the proposed method in the presence of extreme viewpoint differences.

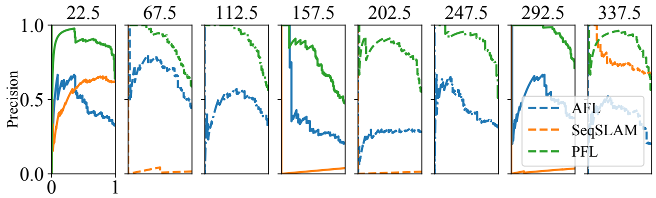
III. Experiments

We use two datasets to test our framework: the *Kitti Odometry* dataset [4] and the *North Campus Long-Term (NCTL)* LiDAR dataset [2]. In both datasets, the bird-view maps are extracted out into image-format based on the proposed DOM module, and resized to 64×64 format. The map scales of DOM are 50m×50m and 30m×30m in the *Kitti* and the *NCTL* respectively. The train and test data samples are generated from difference sequences in each dataset, the detail information is given in Table. I.

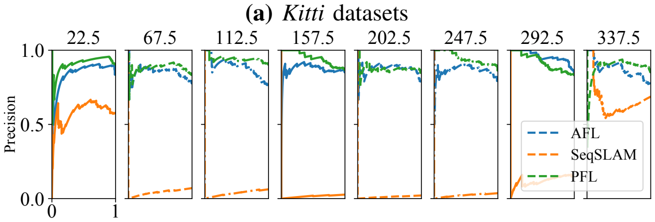
In the experiments, we use the joint 8-bin features to increase the robustness. With the extracted joint PFL feature, the place recognition is then achieved by following the structure of the SeqSLAM approach, where sequences of images' features are matched instead of a single image. For more details about the structure of the SeqSLAM, we refer the readers to read the original SeqSLAM paper [1].

TABLE I: Datasets

Dataset	train	test	map scale
KITTI	27980	2000	50m×50m
NCTL	37890	2000	30m×30m

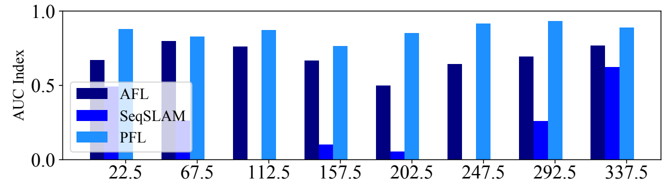


(a) *Kitti* datasets

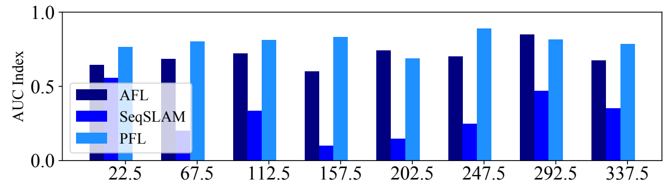


(b) *NCTL* datasets

Fig. 5: Precision-Recall curve of different methods under different heading orientation angles. From the first column to the last column, the heading orientation difference is 22.5° to 337.5° respectively.



(a) AUC on *Kitti* dataset.



(b) AUC on *NCTL* dataset.

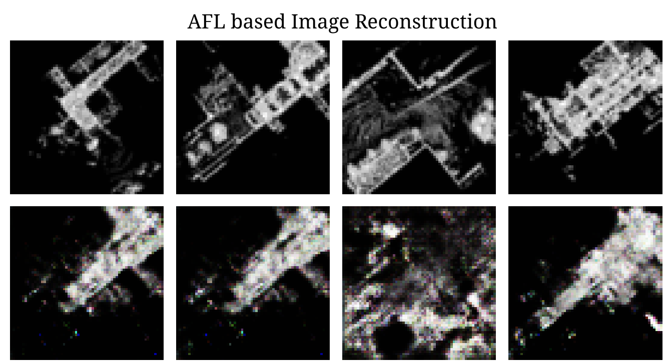
Fig. 6: The AUC index of the place recognition results under different heading orientation situation.

We compare our proposed framework (PFL) with a pure adversarial feature learning method (AFL) [15] and with the original SeqSLAM method that uses the sum of absolute difference (SAD) features. We only focus in these experiments on problems in which we encounter only heading/orientation differences between frames i.e. for the same place, two given frames can only be different in their heading and possibly slight translation. Note that even though there exists other learning-based place recognition methods such as the works in[19], [20], all such methods are either pre-trained on the ImageNet [18] dataset or are based on visual texture features. Thus, these methods cannot be directly compared with our method.

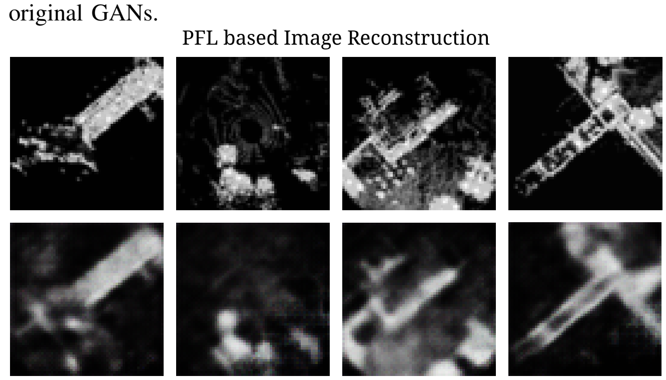
Experiments are conducted on a single Nvidia 1080Ti card with 16G RAM on the Ubuntu 14.04 system. Both of the AFL features and our proposed PFL features are extracted. The SAD feature on the other hand is matched with a fixed angle as in the original SeqSLAM. The rest of this section compares the performance of these various methods in terms of matching accuracy, image recognition, inference time, and storage analysis.

A. Place Recognition Accuracy Analysis

Fig. 5 shows the precision-recall curves for different methods on the *Kitti* datasets and the *NCTL* datasets under different headings. The original SeqSLAM method fails to be effective when the relative heading is bigger than 30 degree. All the methods perform better on the *NCTL* dataset than on the *Kitti* dataset, this is mainly because the scale of the maps in the Kitti dataset is larger than *NCTL*, which introduces more complex geometry information. Compared to other methods, our proposed PFL has a higher recognition accuracy, this is mainly because of the additional GAN module that we add which improves the generalization ability even with limited data samples. Due to the space limitation, we select three constant heading conditions for visualization. Fig. 6 shows the respective AUC indexes which and clearly



(a) In AFL case, different scenes may easily mapped into similar scenes, this is caused by the model collapse problem as in the original GANs.



(b) In PFL case, the global geometry details are kept under the inverse mapping. This enhance the uniqueness from data space to the relative latent code space.

Fig. 7: Scene Reconstruction with $\hat{x} \sim De(En(X))$ based inverse mapping.

reflects the robustness of our PFL method to changes in the heading.

B. Image Reconstruction Analysis

In order to check whether our framework has learned the geometry features irrespective of the local DOM map, we perform a scene reconstruction using both our PFL method and AFL method. This task consist basically of reconstructing a scene with an inverse encoder-decoder mapping. As we can see in Fig. 7, for the AFL method, the reconstructed maps sometimes lose the original geometry and are mapped to similar scenes, which reduces the unique mapping from the DOM map space to the relative latent code space. In our method, the global geometry structure is maintained which improves the unique mapping performance for the DOM maps.

IV. CONCLUSIONS

In this paper, we propose an end-to-end LiDAR-based feature learning framework for long-term place recognition task, where the place recognition is achieved by low dimension feature matching instead of geometry matching. The proposed method is combined with two core modules, a dynamic octree mapping module which generates bird's view of local place with the consideration of the robot's motion, and a place feature inference module that capture unique map-feature mapping with limited data samples. More specially, in place feature learning, we stabilize the feature learning process in an fully unsupervised manner. The experiments conducted on the *Kitti* and the *North Campus Long-Term* LiDAR dataset show that the proposed framework outperforms existing state-of-art methods under variants viewpoints difference.

In future works, we plan to further investigate the multi-bin feature combination mechanisms to achieve more robust heading-invariant features, and combine the proposed place recognition method with the graph-factor approach in order to achieve a global optimization for long-term SLAM framework.

REFERENCES

[1] Niko Sünderhauf, Peer Neubert, and Peter Protzel. "Are we there yet? Challenging SeqSLAM on a 3000 km journey across all four seasons." Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA), 2013.

[2] Nicholas Carlevaris-Bianco, Arash K. Ushani, and Ryan M. Eustice. "University of Michigan North Campus long-term vision and lidar dataset." The International Journal of Robotics Research, vol. 35, no. 9, pp. 1023-1035, 2016.

[3] Ji Zhang and Sanjiv Singh. "LOAM: Lidar odometry and mapping in real-time." Robotics: Science and Systems, vol. 2, pp. 9-17, 2014.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the KITTI vision benchmark suite." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3354-3361, 2012.

[5] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. "Real-time loop closure in 2D LIDAR SLAM." IEEE International Conference on Robotics and Automation (ICRA), pp. 1271-1278, 2016.

[6] Joscha Fossel, Karl Tuyls, Benjamin Schnieders, Daniel Claes, and Daniel Hennes. "NOctoSLAM: Fast octree surface normal mapping and registration." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6764-6769, 2017.

[7] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. "SegMatch: Segment based place recognition in 3D point clouds." IEEE International Conference on Robotics and Automation (ICRA), pp. 5266-5272, 2017.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision (ECCV), pp. 404-417, 2006.

[9] Pauline C. Ng and Steven Henikoff. "SIFT: Predicting amino acid changes that affect protein function." Nucleic acids research, vol. 31, no. 13, pp. 3812-3814, 2003.

[10] David Filliat. "A visual bag of words method for interactive qualitative localization and mapping." IEEE International Conference on Robotics and Automation (ICRA), pp. 3921-3926, 2007.

[11] 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, vol. 32, no. 1, pp. 1-19, 2016.

[12] Mark Cummins and Paul Newman. "FAB-MAP: Probabilistic localization and mapping in the space of appearance." The International Journal of Robotics Research, vol. 27, no. 6, pp. 647-665, 2008.

[13] Michael J. Milford and Gordon. F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights." IEEE International Conference on Robotics and Automation (ICRA), pp. 1643-1649, 2012.

[14] P. J. Besl and Neil D. McKay. "Method for registration of 3-D shapes." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

[15] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning." arXiv preprint arXiv:1605.09782, 2016.

[16] Ian Goodfellow, et al. "Generative adversarial nets." Advances in Neural Information Processing Systems (NIPS), pp. 2672-2680, 2014.

[17] Arjovsky Martin and Léon Bottou. "Towards principled methods for training generative adversarial networks." arXiv preprint arXiv:1701.04862. 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105, 2012.

[19] Sourav Garg, Adam Jacobson, Swagat Kumar, and Michael Milford. "Improving condition- and environment-invariant place recognition with semantic place categorization." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6863-6870, 2017.

[20] Yasir Latif, Ravi Garg, Michael Milford, and Ian Reid. "Addressing challenging place recognition tasks using generative adversarial networks." arXiv preprint arXiv:1709.08810, 2017.

[21] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees." Autonomous Robots, vol. 34, no. 3 pp. 189-206, 2013.

[22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.