IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 9, 2020, accepted April 30, 2020, date of publication June 3, 2020, date of current version June 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999727

LPD-AE: Latent Space Representation of Large-Scale 3D Point Cloud

CHUANZHE SUO[1], **ZHE LIU**[2], **LINGFEI MO**[1], **(Member, IEEE), AND YUNHUI LIU**[3], **(Fellow, IEEE)**
[1]School of Instrumental Science and Technology, Southeast University, Nanjing 210037, China
[2]Department of Computer Science and Technology, University of Cambridge, Cambridge CB2 1TN, U.K.
[3]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong

Corresponding author: Lingfei Mo (lfmo@ seu.edu.cn)

This work was supported by the Key Projects of the National Science and Technology Support Program of China under Grant 2015BAG09B01.

ABSTRACT The effective latent space representation of point cloud provides a foremost and fundamental manner that can be used for challenging tasks, including point cloud based place recognition and reconstruction, especially in large-scale dynamic environments. In this paper, we present a novel deep neural network, LPD-AE(Large-scale Place Description AutoEncoder Network), to obtain meaningful local and contextual features for the generation of latent space from 3D point cloud directly. The encoder network constructs the discriminative global descriptors to realize high accuracy and robust place recognition, which contributed by extracting the local neighbor geometric features and aggregating neighborhood relationships both in feature space and physical space. The decoder network performs hierarchical reconstruction on coarse key points and ultimately produce dense point clouds, which shows that it is capable of reconstructing a full point cloud frame from a single compact but high dimensional descriptor. Our proposed network demonstrates performance that is comparable to the state-of-the-art approaches. With the benefit of the LPD-AE, many computationally complex tasks that rely directly on point clouds can be effortlessly conducted on latent space with lower memory costs, such as relocalization, loop closure detection, and map compression reconstruction. Comprehensive validations on Oxford RobotCar dataset, KITTI dataset, and our freshly collected dataset, which contains multiple trials of repeated routes in different weather and at different times, manifest its potency for real robotic and self-driving implementation. The source code is available at https://github.com/Suoivy/LPD-AE.

INDEX TERMS Point cloud, latent space, place recognition, point cloud reconstruction.

I. INTRODUCTION

LIDAR significantly boosts the progress of self-driving and robotic technologies and becomes the primary sensor to sense the environment for increasing technology maturity and decreasing cost. It can directly depict the real physical world in point clouds, containing real scale measurements and geometric features, which has natural advantages in SLAM(Simultaneous Localization And Mapping) [1]. However, mainstream strategies accomplish loop closure and relocalization tasks with the assistance of camera or GPS [2]–[4], due to the sparsity of laser data, the computational complexity, the absence of effective feature extraction and limited generalization, leaving the promising solution conducted directly on point cloud as an open issue.

Effective latent space representation of point cloud provides a reliable solution, which represents the point cloud by a single global feature utilized to place recognition for solving loop closure and relocalization tasks. To this end, deep learning on the 3D point cloud affords a powerful tool because of its excellent performance on feature extraction and generalization ability. As a pioneer of neural network feature extraction, PointNet [5] laid the foundation of deep learning on the point cloud by applying a symmetric function to each point independently. The improved PointNet++ [6] and subsequent DGCNN [7] introduced neighborhood rather than individual points through hierarchical sampling and dynamic graph network, respectively, to better collect

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

geometric features of the point cloud. Although the above networks have achieved acceptable classification and segmentation results on the ModelNet dataset [8], a single object point cloud dataset generated by CAD model, they can not act as feasible solutions conducted on sophisticated and partial scenes. Designed for the place recognition task, PointNetVLAD [9] combined PointNet and NetVLAD [10], and even PCAN [11] utilized the attention mechanism to reweight features of different point clouds, they both did not consider the uneven density of point cloud, the geometric relationships among points, the different perceptive filed of neighborhood features, and the spatial distribution of local features. What's more, they only paid attention to the generation of discriminative global descriptors but neglected whether it extracted presumptive point cloud information, for lack of reconstruction from global descriptors to corresponding point clouds. In [12], we have proposed LPD-Net, the state-of-the-art place recognition model base on point clouds. Although it exhibits the convincing feature extraction performance for place retrieval, it obtains low similarity of features extracted from similar places for the absence of decoder to recovery point clouds from global features during training.

Unlike the usual reconstruction task that utilizes discrete frames of point clouds or range images to patch up the whole scene [13]–[15], it means the compression reconstruction performed on latent space. FoldingNet [16] proposed the folding operation to construct point clouds surface from codeword produced by the bottleneck layer of the auto-encoder, but it's not suitable for the partial and disconnected large-scale point cloud without modification. SO-Net [17] took advantage of auto-encoder as pre-training to improve the performance of the proposed self-organized map on the ModelNet dataset. SegMap [18] leveraged a data-driven descriptor to extract the feature of voxelized segments in point clouds and performed reconstruction with 3D convolutional layers, which cost the amount of computation and couldn't adapt to the whole large-scale scene due to the sparsity. The upsampling networks can be applied to the large-scale scene and generate more dense point clouds, but they utilize more detailed point-wise features rather than global features, so these are not in the scope of this paper. As an essential chain of the transformation between the latent space and the point space, reconstruction indicates the complete mapping between them. It shows the possibility to replace thousands of points with a 256-dimensional vector for point cloud operations. By virtue of the latent space completeness, it brings forth potency that complex computational tasks, such as loop closure, relocalization, and compression reconstruction, can be conducted on latent space with limited memory, communication bandwidth, and computing resource.

To address the above issues, we present a novel and complete latent space representation pipeline, LPD-AE (Large-scale Place Description AutoEncoder Network), which consists of the following two parts for recognition and reconstruction:

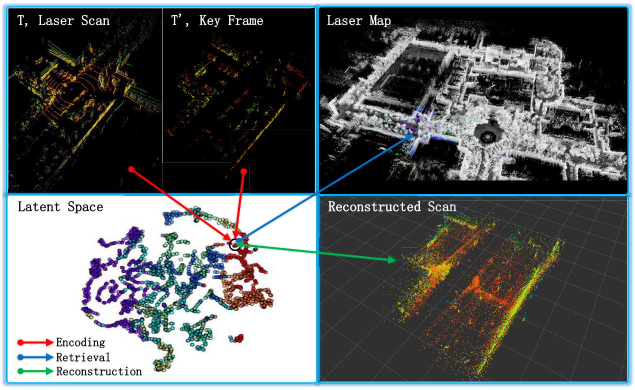

FIGURE 1. Latent space representation for large-scale point clouds. The top-left image shows the current laser scan and downsampled keyframe point cloud of the same place. In latent space depicted by t-SNE in the bottom-left image, the encoder network generates embeddings with a closer distance for similar point clouds, to conduct loop closure or relocalization task by retrieval in laser map shown in the top-right image. The reconstructed scan produced by the decoder network from the corresponding embedding feature is shown in the bottom-right image.

1) The encoding architecture, constructing the latent space to represent the point cloud space, produce discriminative global descriptors from point clouds directly by aggregating low-level geometric features and high-level context features, which same to the LPD-Net [12]. An adaptive local feature extraction module is leveraged to organize the neighborhood of each point adaptively. The graph-based aggregation in feature space and physical space can reveal the context features and their spatial distribution cue, which also makes the receptive filed of high-dimensional features breakthrough the scope of a single geometric structure or single instance.

2) The decoder architecture, reconstructing the corresponding point cloud from compressed global descriptors, confirms the validity and completeness of the latent space representation. Through the hierarchical generation strategy, coarse key points are produced by the efficient but straightforward fully connected layers, which get rid of the challenges from the irregular structure of point clouds and the computational cost from the 3D CNN applying to voxels [19]. And then, segmented mesh grid constraint is introduced to generate dense point clouds by the folding-like operation [16]. The lightweight network is designed, yet lead to a reconstruction that thoroughly mines information in the global descriptor and avoids too deep network memory irrelevant data. All of it is achieved by supervision according to the position and distribution of the point cloud.

We summarize the key contributions as follows:

- We propose a novel and effective latent space representation for large-scale place descriptions (Fig.1), leveraging the LPD-AE network to conduct some point cloud-related tasks on latent space, such as loop closure, relocalization, and compression reconstruction task.
- To the best of our knowledge, this is the first work on the reconstruction of the large-scale scene from a single

descriptor for place retrieval, which shows an exciting capability for the reconstruction of the real-world point cloud with lower memory and bandwidth cost.
- In addition to retaining the discrimination of LPD-Net encoding features, we have improved the similarity of global descriptor in similar scenes.
- We demonstrate comprehensive validation and application potency on different datasets, including the freshly collected dataset with multiple trials of repeated routes in different weather and at different times.

The paper is organized as follows. In Section II, related works about deep learning on the point cloud, place recognition and point cloud reconstruction are introduced. Section III presents the system structure and the statement of the problem. Components of the proposed LPD-AE network are well defined in Section IV. Extensive experiments and detailed analysis are clarified in Section V. Section VI demonstrates comprehensive applications with qualitative results, which is followed by the conclusion in Section VII.

II. RELATED WORKS

A. DEEP LEARNING ON POINT CLOUD

The development of image processing has matured, including traditional manual feature design [20]–[22] and various models [23], [24] based on deep learning. However, 3D point cloud processing is still a very challenging problem because of the irregular, unordered, and rotation invariance of point clouds. In the early years, many handcraft features based on geometric or statistical are still powerful description methods beyond deep learning, including Spin Images [25], Geometry Histograms [26], Point Feature Histograms (PFH) [27], Fast Point Feature Histograms (FPFH) [28], and Signature of Histogram Orientations (SHOT) [29].

Deep learning has an excellent performance in extracting features and has taken the place of many traditional description methods. In the early stages of the development, Some researchers attempt to transform irregular point cloud data into regular data representations, such as projection view and volume. Some works [30], [31] refer to deep learning models applying in images by projecting 3D objects into multi-views and utilizing standard convolution operations to extract features. Equivalent to the concept of pixels, it is also a very straightforward method to voxelized 3d objects into regular data. Voxelization [8], [19], [32] provides a way to index and organize data, but because of the sparseness of voxelization, applying standard 3D CNNs is very wasteful. Based on this method, the required storage capacity and computational complexity are limited by the resolution of voxelization. Subsequently, for solving the resolution problem, KDtree and Octree [33]–[35] are applied but still depended on the minimum resolution of volume segmentation.

PointNet [5] is a pioneering work that directly processes point cloud data with symmetric MLP operations that are applied to each point to extract features. However, its performance is limited for lacking point cloud neighborhood. The improved version, PointNet++ [6] extract hierarchical neighborhood information by ball queries and FPS operation. DGCNN [7] uses a dynamic graph network to adjust neighborhood relationships in a data-driven manner, which is more reasonable. PointCNN [36] proposes the $\mathcal{X}$-transformation module, which uses dynamic graphs on transformed feature space to solve the permutation invariant problem. KPConv [37] proposes an innovative convolution, Kernel Point Convolution, which can be used directly on the point cloud without any intermediate representation, which solves the problem that KNN is not robust on the non-uniform sampling point cloud.

More and more deep learning networks are designed to extract point cloud geometric features in a data-driven manner, but traditional strictly mathematical description features [38] are still very active. Especially in the complex and large-scale point cloud, it is difficult for neural networks to extract meaningful features.

B. PLACE RECOGNITION

Given a query pictures or point clouds, the best match is retrieved in the database according to the descriptor generated by place recognition methods [3], [4], [39]–[41]. Place recognition can determine whether scenes or parts of scenes are the same matches. The most common application scenarios are relocalization and closed-loop detection tasks in SLAM. Efficient and distinguishing feature description methods are the key to solve this problem, which is mainly divided into image-based and point cloud-based.

Robust image feature methods dominate place recognition because the image contains rich scene information. The SIFT [22] descriptor with local invariance is the most commonly used method, which extracts features of the picture, partly or totally. With the aggregation of these features, a compressed and efficient index is produced through a bag of words model [42], [43], VLAD [44], [45], or Fisher vector [46], [47]. There are many improved designs [4], [48]–[51] in image retrieval and place recognition in order to better solve this problem, but many are based on traditional methods rather than learning-based methods. Among the many learning-based methods, including local feature learning, metric improvement, and CNN features, NetVLAD [10] provides an end-to-end solution. It uses VGG/AlexNet [23], [52] to extract features, followed by a trainable generalized NetVLAD layer, aggregates local features into a global description vector. Robust description features for place recognition are learned, which are not affected by changes in perspective.

Images have natural shortcomings and are susceptible to changes in viewpoint, lighting, weather, and season. Point cloud data can make up for these problems, but suffering the lack of robust descriptor like SIFT [22] for point clouds. Many practical applications use GPS to provide a rough position, followed by the registration method like ICP-series [53]. SegMatch [41] and SegMap [18] leverage the learning method to extract features of the segmented point cloud, regarding segments without intuitions in the scene as processing units for place recognition. Inspired by NetVLAD, PointNetVLAD [9] combines the features extracted by PointNet with the NetVLAD layer, which is the first point cloud-based place recognition approach using deep learning, and a benchmark for scene recognition based on point clouds is created. However, PointNet cannot extract sufficient features in large-scale point clouds, and the neighborhood relationship was neglected. PCAN [11] introduced the attention mechanism to reweight features and learn more significant features, but still did not consider the above problems.

Our proposed LPD-net [12], as an encoder for latent space representation, has been optimized to meet the needs of the large-scale point cloud, which strengthened the geometric structure and neighborhood relationship to extracted the discriminative global descriptor. SepLPD [54] has verified its practicability and effectiveness in autonomous driving applications. But the previous work lacks an essential component of latent space representation, which is the reconstruction. Comparison with our previous work [12], [54], we significantly improve the discrimination performance of LPD-Net and study the reconstruction of the large-scale scene in this paper for the first time.

C. POINT CLOUD RECONSTRUCTION

Point cloud compression reconstruction is a significant and challenging problem. Most of the research [16], [17], [55], [56] is mainly focus on generating more efficient intermediate codes for classification and constructing latent space representation. There are two main classify for point cloud reconstruction: autoencoders and adversarial generation networks [57]–[59]. This article only focuses on the compression reconstruction of the point cloud. The point clouds completion [60]–[62] and the SfM(Structure from Motion) [63] are not in the scope of discussion.

FoldingNet [16] proposes a compression-reconstructed network using an autoencoder, generates a codeword by a graph-based encoder, and restores the point cloud through the proposed folding operation by squeezing the 2D mesh grid. SO-Net [17] is a similar pipeline, which builds a Self-Organizing Map(SOM) [64] to generate global vectors, with systematically adjustable receptive fields. However, these networks are mainly conducted on the CAD-generated dataset, ModelNet [8], and may not suitably be applied to real-world point clouds. Some network [65] use autoencoders only to generate description vectors for local subsets of points for the registration purpose, not including reconstruction. SegMap [18] also extracts features from segments in the point cloud using an autoencoder, which uses the decoder network to recover local point clouds from features to reconstruct the map. However, these ways of applying reconstruction in large scenes are all delivered to a part of the point cloud, not the entire one.

Adversarial generative networks are a fabulous way to reconstruct point clouds and demonstrate extraordinary capabilities on the ModelNet. The adversarial autoencoder models AE [57] and 3dAAE [58] represent the latent space of the point cloud and show impressive results in feature interpolation and latent space editing. Although conducted on a generated dataset, they provide the immense potential for the latent space representation of large-scale point clouds, which will be the direction of subsequent research.

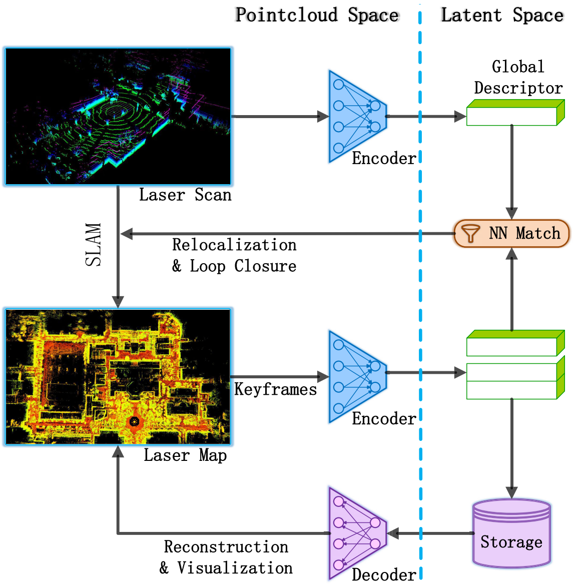

FIGURE 2. System structure shows the pipeline of latent space representation. All it takes is only a 256-dimensional vector representing a whole point cloud in latent space to conduct complex computational tasks, such as loop closure, relocalization, and compression reconstruction, with limited memory, communication bandwidth, and computing resource.

III. SYSTEM STRUCTURE AND PROBLEM STATEMENT

The objective of this paper is to construct a complete latent space representing the point cloud space, using discriminative global descriptors extracted from large-scale point clouds, and based on which to reconstruct the corresponding original point cloud. The ability to bi-directionally transform of two spaces given by the LPD-AE, make it feasible to perform complex computational tasks in point clouds by manipulating latent space vectors, such as classification, loop closure, relocalization and compression storage.

The system structure of this paper is shown in Fig.2. The point cloud $P = \{p_1, \ldots, p_N \mid p_n \in \mathbb{R}^3\}$ of laser scan obtained by the lidar is the system input, and it is the implementation object of mapping and descriptor encoding. On the one hand, in the process of SLAM mapping, the feature extraction of the input point cloud is performed in real-time for inter-frame matching and odometer motion estimation. On the other hand, the global descriptor $f(P) \in \mathbb{R}^\Gamma$ of the current scan is extracted using the proposed encoder network $\mathfrak{E}$: $P \xrightarrow{\mathfrak{E}} f(P)$, mapping to the latent space. The laser map $M = \cup_{\kappa=0}^{\boldsymbol{K}} m_\kappa$ is stored and maintained using keyframes set, which include position t, pose R, and point cloud data P. The global descriptor is also stored as an additional attribute value in the keyframes set $\{m_1, \ldots, m_{\boldsymbol{K}} \mid m_\kappa = \{t_\kappa, R_\kappa, P_\kappa, f(P_\kappa)\}\}$.

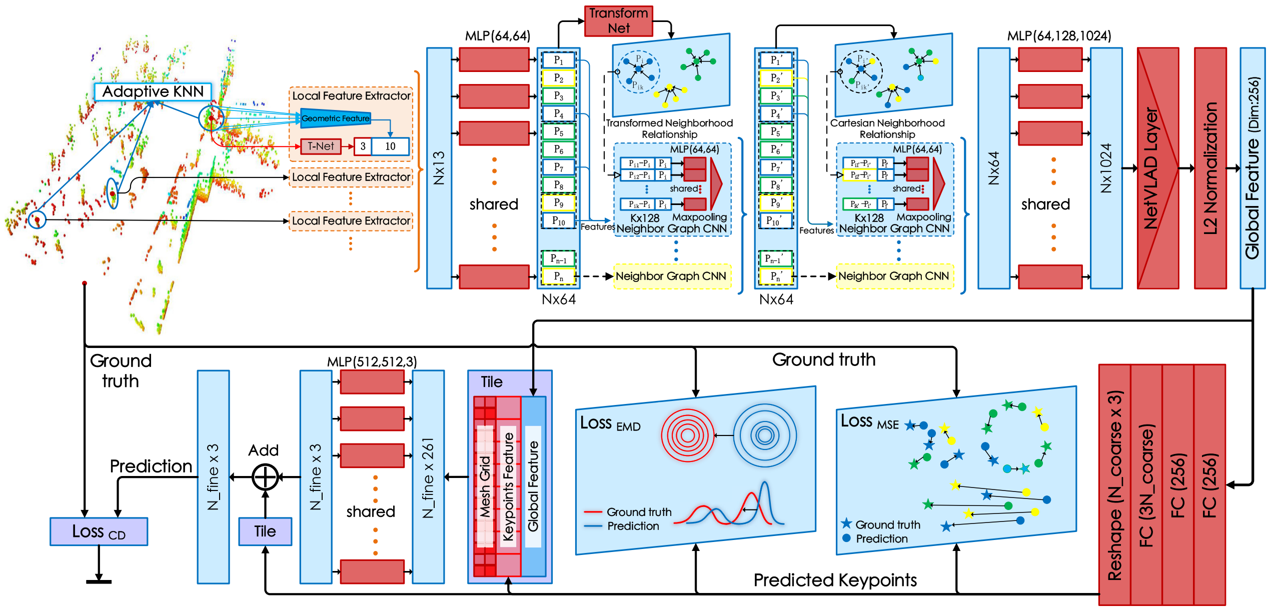

FIGURE 3. Network Architecture of the LPD-AE, consisting of the Decoder network(above) and the Encoder network(below).

For mapping, in addition to using the point cloud for odometer motion estimation, it is also performing loop closure detection. With the latent space representation, the Euclidean distance-based matching strategy $S : d(f(P), f(\bar{P})) < \bar{d}$ is leveraged to determine whether the current point cloud matches the element of the maintained keyframes deposited $\hat{m} = \{\bar{t}, \bar{R}, \bar{P}, f(\bar{P})\} \in M\}$. If matches, the relative pose between the current scan and point cloud of loop closure candidate $\hat{P}$ are estimated to optimize each keyframe's pose and position in the whole map.

In the process of localization, similar operation on global descriptor matching is also performed in the built-map. The maintained keyframe descriptor $f(P) \in \mathbb{R}^{\Gamma}$ dramatically reduces the storage space compared to the original point cloud data $P = \{p_1, \ldots, p_N \mid p_n \in \mathbb{R}^3\}$, where $\Gamma \ll (N \times 3)$, which is convenient for storage and low-bandwidth transmission. The rebuilt laser map $\hat{M} = \cup_{\kappa=0}^{\boldsymbol{K}} \hat{m}_{\kappa} \mid \hat{m}_{\kappa} = \{t_{\kappa}, R_{\kappa}, \hat{P}_{\kappa}, f(P_{\kappa})\}$ is reconstituted from point clouds reconstructed $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_O \mid \hat{p}_o \in \mathbb{R}^3\}$ with global descriptors $f(P)$ through the decoder network $\mathfrak{D} : f(P) \overset{\mathfrak{D}}{\Longrightarrow} \hat{P}$.

We are committed to designing a more solid encoder $\mathfrak{E}$ that maps similar point clouds $P = \{p_1, \ldots, p_N \mid p_n \in \mathbb{R}^3\}$ to latent space with closer global descriptor distances $d(f(P), f(P_s)) < d(f(P), f(P_d))$, where P_s is structurally similar to P and P_d not, and a more suitable decoder $\mathfrak{D}$ to reconstruct the descriptor $f(P)$ into the point cloud $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_O \mid \hat{p}_o \in \mathbb{R}^3\}$ with dense geometry $O > N$ and closer spatial distribution with the original point cloud $\Psi(P, \hat{P}) \to 0$, where $\Psi(.)$ is the metric of distribution.

IV. LPD-AE NETWORK

In this section, we will elaborate on the network structure of the LPD-AE network, including the loss function to supervise the network. All of it is targeting real large-scale point clouds.

A. NETWORK ARCHITECTURE

The overall design structure of the LPD-AE network is shown in the Fig.3. It is a two-stage multi-tasking framework of the autoencoder, which completes the peer-to-peer mapping of point cloud space and feature space. The point cloud of a real large-scale scene is different from the point cloud in a single object data set, e.g., ModelNet and ShapeNet [66]. The large-scale point clouds contain many different objects at different positions, even dynamic objects. Besides, the point clouds of objects are inevitably partial because of the projection principle of laser acquisition. Therefore, The neural networks designing for classical classification are not intuitively applicable, which are suitable for directly learning the features of instance objects. We believe that efficient local structural features and distribution characteristics of them in the scene provide clues for encoding large-scale point clouds, consequently can facilitate representing and reconstructing the original point cloud.

The upper part of the Fig.3 is the point cloud encoder network $\mathfrak{E}$, LPD-net [12]. The input is a normalized and random downsampled point cloud, in the form of $N \times 3$ matrix, each row represents the position of a 3D point (x, y, z). And the output is a $1 \times \Gamma$ vector, which represents the corresponding global descriptor of the point cloud. A semi-supervised learning method based on metric learning [67] is applied to encoder training.

The lower part of the Fig.3 is the decoder network $\mathfrak{D}$ of the point cloud. The input is the $1 \times \Gamma$ vector generated above, and the output is a matrix of $O \times 3$, which represents the reconstructed point cloud. The decoder is trained using supervised learning.

$$\mathcal{LPD}-\mathcal{AE} : \left\{ \underbrace{\overbrace{P \overset{\mathfrak{E}}{\Longrightarrow} f(P)}^{Loss_{recognition}} \overset{\mathfrak{D}}{\Longrightarrow} \hat{P}}_{Loss_{reconstruction}} \right.$$

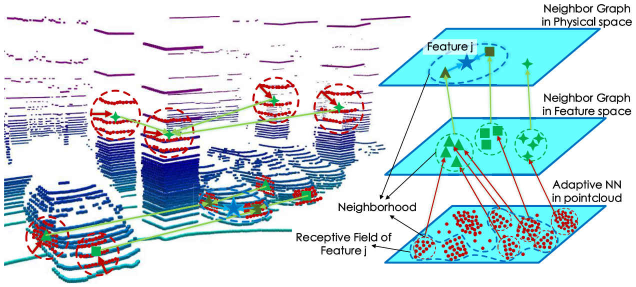
Neighbor Graph in Physical space
Feature j
Neighbor Graph in Feature space
Adaptive NN in pointcloud
Neighborhood
Receptive Field of Feature j

FIGURE 4. **Demonstration of Feature aggregation and Perceptive filed of the feature. The figure depicts the process of feature extraction and aggregation. Through adaptive neighborhood, feature space, and physical space graph aggregation, more representative contextual features can be extracted, and the corresponding receptive field can exceed the scope of a single structure or instance.**

B. ENCODER ARCHITECTURE

Against the issues mentioned above, LPD-Net [12] is adopted as the encoder with the following modules.

Adaptive neighborhood feature extraction is to solve the sparseness and unevenness of point clouds. Objects in large-scale point clouds have different scales, and the point clouds of them tend to be denser as they get closer, and vice versa. These can be solved by determining the optimal size of the neighborhood surrounding the point. And the enhanced geometric features are used to describe the neighborhood.

Dynamic graph feature aggregation is used to obtain more robust and representative low-level structure features and high-level context features. The aggregated features contain semantic information, implying local descriptors and their distribution characteristics in space. Through this module, the receptive field of high-level features can be adjusted systematically beyond the scope of a single structure or a single object, as depicted in Fig.4.

The trainable generalized Vector of Locally Aggregated Descriptors (VLAD) layer, NetVLAD layer, is applied to aggregate the local features into a global descriptor. Then semi-supervised learning is conducted through the $\mathcal{L}_{recog*}$.

Each module of the encoder neural network is elaborated as follows.

1) ADAPTIVE NEIGHBORHOOD FEATURE EXTRACTION

Given a point cloud $P \subseteq \mathbb{R}^3$, in the form of $N \times 3$ matrix, we construct a neighborhood $\mathcal{N}_i \subseteq \mathbb{R}^3$ with size of $k \times 3$ for each point p_i with an adaptive neighborhood size k-nearest neighbor (kNN) to represent the local structure of the point cloud, by k ranging from $[k_{min}, k_{max}]$. The respective 3D covariance matrix Σ of neighborhoods is considered to be a local structure tensor with three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and corresponding orthogonal eigenvectors $V = (V_1, V_2, V_3)$ because it is a symmetric positive definite matrix, considering the general structure tensor with rank 3, according to

$$\Sigma = Cov(\mathcal{N}, \mathcal{N}) = Vdiag(\lambda_1, \lambda_2, \lambda_3)V^T \tag{1}$$

According to [38], Eigenvalue-based feature of linearity L_λ, planarity P_λ, and scattering S_λ according to

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}, \quad P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}, \; S_\lambda = \frac{\lambda_3}{\lambda_1} \tag{2}$$

represent the 1D, 2D, and 3D features of local structures, which are used to construct a measurement E_k to describe unpredictability given by the Shannon entropy as

$$E_k = -L_\lambda \ln L_\lambda - P_\lambda \ln P_\lambda - S_\lambda \ln S_\lambda, \tag{3}$$

Under the condition that the point distribution in the point cloud is typically uniform, the optimal neighborhood size k_{opt} of each point p_i can be determined adaptively through E_k.

$$k^i_{opt} = \arg\min_k E_(k). \tag{4}$$

Various features in [12], [38], [54] can be extracted from the local structure, and we chose to describe the local structure of each point p_i with features based on 3D eigenvalues, 2D eigenvalues, and 1D Z-direction distribution.

- Linearity: $L = \frac{\lambda_1 - \lambda_2}{\lambda_1}$
- Eigenvalue-entropy: $A = -\sum_{j=1}^{3}(\lambda_j \ln \lambda_j)$
- Change of curvature: $C = \frac{\lambda_3}{\sum_{j=1}^{3}\lambda_j}$
- Omni-variance: $O = \frac{\sqrt[3]{\prod_{j=1}^{3}\lambda_j}}{\sum_{j=1}^{3}\lambda_j}$
- Local point density: $D = \frac{k_{opt}}{\frac{4}{3}\prod_{j=1}^{3}\lambda_j}$
- 2D scattering: $S_{2D} = \lambda_{2D,1} + \lambda_{2D,2}$
- 2D linearity: $L_{2D} = \frac{\lambda_{2D,2}}{\lambda_{2D,1}}$
- Vertical component of normal vector: V
- Maximum height difference: $\Delta Z_{\max}$
- Height variance: σZ_{var}

Denote the neighborhood features of the point cloud by $\mathcal{N}_f \subseteq \mathbb{R}^{10}$, and each row is composed of the normalized features$(L, A, C, O, D, S_{2D}, L_{2D}, V, \Delta Z_{\max}, \sigma Z_{var})$.

The learnable T-Net [5], [68] can facilitate the feature extracted from the point cloud to obtain the invariance to transformations, i.e., $P' = PA_{(T,\theta)}$, where $A_{(T,\theta)}$ is a 3×3 matrix generated by T-Net with parameters θ. Thus, each row of P' is a transformed 3D position(x', y', z'). Then, extend the original point cloud with the transformed point cloud and the neighborhood features, i.e., $P \subseteq \mathbb{R}^3 \mapsto \{P' \| \mathcal{N}_f\} \subseteq \mathbb{R}^{13}$, where $\|$ is concatenation operation over feature channels.

Furthermore, with the benefits of neural network, the neighborhood features can better learn by mapping to higher-dimensional space by a parametric non-linear function, i.e., $\{P' \| \mathcal{N}_f\} \subseteq \mathbb{R}^{13} \mapsto \mathcal{F} = \{f_1, \ldots, f_N \mid f_n \in \mathbb{R}^F\}$, with F-dimension.

2) DYNAMIC GRAPH FEATURE AGGREGATION

Considering point clouds $\mathcal{F} \subseteq \mathbb{R}^F$ with local structural features, we further assume that directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ is given to aggregate local neighborhood structure features, where $\boldsymbol{V} = \{1, \ldots, N\}$ and $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ are the vertices for nodes of points and edges for neighboring pairs of points in the form $(i, j_{i1}), \ldots, (i, j_{ik})$, respectively.

To ensure that the edge relationship of the constructed graph can fit the invariance of feature transformation, We adopt a learnable feature transformation matrix $A \in \mathbb{R}^{F \times F}$ predicted by a mini-network according to

$$A = \boldsymbol{l}_{\Theta}(\boldsymbol{g}_{\Theta}(f_i)), \quad \forall f_i \in \mathcal{F} \tag{5}$$

where $\boldsymbol{l}_{\Theta}$ and $\boldsymbol{g}_{\Theta}$ are parametric non-linear function with learnable parameters $\theta_i \in \Theta$, $\boldsymbol{g}_{\Theta} : \mathbb{R}^F \times \mathbb{R}^F \mapsto \mathbb{R}^{F'}$, and $\boldsymbol{l}_{\Theta} : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \mapsto \mathbb{R}^{F \times F}$, which constrained to be close to the orthogonal matrix by regularization term aiming not to lose information:

$$L_{reg} = \|I - AA^T\|_F^2 \tag{6}$$

Thus, the invariance of feature transformation is satisfied by $\mathcal{F}' = \mathcal{F}A$ for neighboring edge relationship.

Local structure feature aggregation is determined by kNN with fixed neighborhood size $k_{\mathcal{F}}$ in the feature space, where vertices denoted as $\boldsymbol{V}_{(\mathcal{F})} \subseteq \mathcal{F}$, and edges relationship $\boldsymbol{E}_{(\mathcal{F}')}$ is constructed from transformed feature space$(F)'$:

$$\begin{aligned} \boldsymbol{E}_{(\mathcal{F}')} &= kNN(f_j' \mid f_i', j = 0, \ldots N, f_i', f_j' \in \mathcal{F}') \\ &= \{(i, j_{i1}), \ldots, (i, j_{ik}) \mid k = k_{\mathcal{F}}\} \end{aligned} \tag{7}$$

where (i, j_{ik}) represents pointf_{ik}' is one of the k nearest points of pointf_i'.

Aggregate and update the edges and vertices of the graph [69] to produce the F_{l+1}-dimensional output of the $(l + 1)$st layer by applying to the F_l-dimensional output of the l-th layer:

$$\begin{aligned} f_i^{(l+1)} &= \boldsymbol{\rho}^{e \to v}(\boldsymbol{\phi}_{\Theta}^{e}(\mathcal{E}(f_i^{(l)}, f_j^{(l)}))) \\ &= \boldsymbol{\rho}^{e \to v}(\boldsymbol{\phi}_{\Theta}^{e}(f_i^{(l)} \| (f_j^{(l)} - f_i^{(l)}))) \\ &\quad \forall e : (i, j) \in \boldsymbol{E}_{(\mathcal{F}')}, \forall v : f_{\Box} \in \boldsymbol{V}_{(\mathcal{F})} \end{aligned} \tag{8}$$

where $\mathcal{E}(f_i, f_j)$ is the feature between the point f_i and f_j, $\|$ is concatenation operation, $\boldsymbol{\phi}_{\Theta}^{e} : \mathbb{R}^{F_l} \times \mathbb{R}^{F_l} \mapsto \mathbb{R}^{F_{l+1}}$ is a learnable non-linear aggregation function to represent edge features in the form of $k_{\mathcal{F}} \times F_{l+1}$ matrix, and $\boldsymbol{\rho}^{e \to v} : [k_{\mathcal{F}} \times F_{l+1}] \mapsto [1 \times F_{l+1}]$ is to update the feature of vertex v with edge features. Denote the output of aggregation in the feature space by $\mathcal{F}_f \subseteq \mathbb{R}^{F_f}$.

Context features reveal the distribution of structures around points in physical space $\mathcal{C}$, which are organized using kNN with fixed size $k_{\mathcal{C}}$. Please note that we construct the edges $\boldsymbol{E}_{(\mathcal{C})}$ by the corresponding original 3D position in point cloud P and utilize features of $\mathcal{F}_f$ to represent the vertices $\boldsymbol{V}_{(\mathcal{F}_f)} \subseteq \mathcal{F}_f$:

$$\begin{aligned} \boldsymbol{E}_{(\mathcal{C})} &= kNN(p_j \mid p_i, j = 0, \ldots N, p_i, p_j \in P) \\ &= \{(i, j_{i1}), \ldots, (i, j_{ik}) \mid k = k_{\mathcal{C}}\} \end{aligned} \tag{9}$$

Similar to (8) but performed with $\boldsymbol{E}_{(\mathcal{C})}$ and $\boldsymbol{V}_{(\mathcal{F}_f)}$, through the graph-based aggregation in physical space, context features can be obtained as $\mathcal{F}_{\mathcal{C}} \subseteq \mathbb{R}^{F_c}$, a $N \times F_c$ matrix.

3) GLOBAL DESCRIPTOR AGGREGATION

The NetVLAD initiates it feasible to generate the global descriptor aggregating local features through a learnable network. Given a set of N F_c-dimensional local features, i.e., $\mathcal{F}_{\mathcal{C}} = \{f_{c1}, \ldots, f_{cN} \mid f_{ci} \in \mathbb{R}^{F_c}\}$, NetVLAD layer produces a descriptor matrix, V with $D_v \times K_v$ dimension, by learning K_v D-dimensional clustering centers $\{c_1, \ldots, c_{K_v} \mid c_{k_v} \in \mathbb{R}^{D_v}\}$ and weighting residuals $(f_{ci} - c_{k_v})$ of local features f_{ci} and clustering centers c_{k_v} by the adjustable soft-assignment $\mathcal{A}_{k_v}(f_{ci})$:

$$\begin{aligned} V(j, k_v) &= \sum_{i=1}^{N} \mathcal{A}_{k_v}(f_{ci}) \cdot (f_{ci}(j) - c_{k_v}(j)) \\ &= \sum_{i=1}^{N} \mathcal{S}(\boldsymbol{w}_{k_v}^T f_{ci} + \boldsymbol{b}_{k_v})(f_{ci}(j) - c_{k_v}(j)) \end{aligned} \tag{10}$$

where $\mathcal{S}(\cdot)$ is the Softmax function, weights $\boldsymbol{w}_{k_v}$, $\boldsymbol{b}_{k_v}$ and cluster centers c_{k_v} are learned by the network more flexibly. Considering that the output of the NetVLAD layer is a $D_v \times K_v$ matrix, the fully connected layer $\mathbb{R}^{D_v \times K_v} \mapsto \mathbb{R}^{\Gamma}$ and the following $L2$-normalization are applied to extract the final global description vector $f(P) \in \mathbb{R}^{\Gamma}$ and $\|f(P)\|_2 = 1$.

C. DECODER ARCHITECTURE

To recover and reconstruct the point cloud from the global descriptor generated by the above encoder, we use a hierarchical coarse-to-fine reconstruction strategy, from the rough sketch to the detailed structure, which shows in the lower part of Fig.3.

Based on different distance measurements, constrain the distribution and position of the reconstructed point cloud, with $\mathcal{L}_{recons*}$.

The specific structure of the decoder is introduced as follows.

1) BACKBONE NETWORK OF THE ENCODER

In order to learn the scene information hidden in the global descriptor thoroughly, a lightweight network is applied in the decoder. Otherwise, heavy networks tend to learn irrelevant information of the dataset and make the network memorize the data instead of reasoning the peer-to-peer mapping between the point cloud and descriptor.

Compared with reconstruction using the 3D CNN in the form of voxels, a fully connected network can save computation and memory. However, directly generating a $3 \times O$ vector still requires heavy memory and computation if the number of points O and the number of layers is large.

Therefore, we choose hierarchical, coarse-to-fine generation strategy, with the straightforward and effectively fully connected layer and multilayer perceptron, which not only saves computation and memory but also suffices the intuition of generating point clouds.

2) HIERARCHICAL RECONSTRUCTION

The hierarchical generation strategy is to first generate the key points of the rough sketch through the fully connected network, which conforms to the spatial distribution of the point cloud. Then cover a 2D mesh grid around the key points and perform folding-like operations to fit the fine structure for generating a dense point cloud.

Given a $1 \times \Gamma$ matrix that represents the global descriptor$f(P) \in \mathbb{R}^{\Gamma}$, first expand the dimension to extract the information through a parametric non-linear function h_{Θ}, and map the expanded vector to a set P_{coarse} in the form of $N_c \times 3$ through the function ∇ to represent the initially generated point cloud sketch:

$$P_c = \nabla\, h_{\Theta}(f(P)) \tag{11}$$

where $h_{\Theta} : \mathbb{R}^{\Gamma} \times \mathbb{R}^{\Gamma} \mapsto \mathbb{R}^{3N_c}$, ∇ is a mapping function.

Then consider the key point $p_{ci} \in P_c$ of the rough sketch, generate a 2D mesh grid $\mathcal{M}(\eta, \xi) \subseteq \mathbb{R}^{\eta \times \eta}$ around the point according to:

$$\begin{aligned}\mathcal{M}(\eta, \xi) = \{(\boldsymbol{m}_{xi}, \boldsymbol{m}_{yi}) \mid \boldsymbol{m}_{xi}, \boldsymbol{m}_{yi} = i\frac{2\xi}{\eta - 1} - \xi\}\\ \forall \boldsymbol{m}_{xi}, \quad \forall \boldsymbol{m}_{yi} \in [-\xi, \xi],\ \eta > 1\end{aligned} \tag{12}$$

and perform the *Folding* operation [16] on the mesh grid to construct a fine local structure $\mathcal{S}_i \subseteq \mathbb{R}^3$ in terms of the global vector and the point p_{ci}:

$$\begin{aligned}\mathcal{S}_i &= Folding(\mathcal{M}(\eta, \xi), p_{ci}, f(P))\\ &= \sigma_{\Theta}(\mathcal{M}(\eta, \xi) \parallel p_{ci} \parallel f(P))\end{aligned} \tag{13}$$

where σ_{Θ} is also a learnable non-linear function, and $\parallel$ is a concatenation operation. With (13), the mesh grid can be deformed to the real surface $\hat{\mathcal{S}}$ around the keypoint.

Therefore, the *folded* mesh grids are added to the key points and stitched the reconstructed point cloud:

$$\hat{P} = \cup_{i=0}^{N} (\mathcal{S}_i + p_{ci}) \tag{14}$$

where the $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_O \mid \hat{p}_o \in \mathbb{R}^3\}$ is the reconstruction point cloud, with the form of $O \times 3$ matrix.

D. LOSS FUNCTIONS

Similar to PointNetVLAD [9], The lazy quadruplet loss based $\mathcal{L}_{lazyQuad}$ on metric learning is utilized in the place recognition loss $\mathcal{L}_{recog*}$, which ensures that the encoder generates discriminative global descriptors so that similar point clouds have a closer distance in the feature space.

In order to supervise the reconstruction network, the loss function $\mathcal{L}_{recons*}$ for the spatial distribution and position constraints on the point cloud is also used, including the MSE loss $\mathcal{L}_{MSE}$, Chamfer Distance loss $\mathcal{L}_{CD}$ and Earth Mover's Distance loss $\mathcal{L}_{EMD}$ [15].

1) METRIC LEARNING LOSS

Considering a tuple $\mathcal{T} = (P_t, P_{pos}, \{P_{neg}\})$ of point clouds for training, including the target point cloud P_t, a similar point clouds P_{pos}, and dissimilar point clouds $\{P_{neg}\}$, the optimization goal is to reduce the global descriptors' distance between the positive matches $\delta_{pos} = d(f(P_t), f(P_{pos}))$, and enlarge the distances between the negative matches $\{\delta_{neg} = d(f(P_t), f(P_{neg}))\}$. To avoid the confusion of two dissimilar point clouds, i.e., P_{neg} and P_{neg*}, enlarge the distance of them i.e., $\{\delta_{neg*} = d(f(neg*), f(P_{neg}))\}$ for stable training, P_{neg*} is randomly selected during the training process. The lazy quadruplet loss is utilized as:

$$\begin{aligned}\mathcal{L}_{lazyQuad} = \max([\alpha + \delta_{pos} - \{\delta_{neg}\}]_{+})\\ + \max([\beta + \delta_{pos} - \{\delta_{neg*}\}]+)\end{aligned} \tag{15}$$

where α and β are constant margin parameters.

2) RECONSTRUCTION LOSS

Since the disorder of the point cloud, it is challenging to learn the pairing relationship between the reconstructed point cloud $\hat{P}$ and the ground truth point cloud P. The solutions can be summarized into two categories.

One is a loss function that satisfies the invariance to permutations, e.g., Chamfer Distance loss $\mathcal{L}_{CD}$ in (16) and Earth Mover's Distance loss $\mathcal{L}_{EMD}$ in (17), determining the matching relationship through nearest neighbor searching, and the network learns the sequence pattern of output systematically.

$$\begin{aligned}\mathcal{L}_{CD}(P, \hat{P}) = \frac{1}{|P|}\sum_{p \in P} \min_{\hat{p} \in \hat{P}} \|p - \hat{p}\|_2\\ + \frac{1}{|\hat{P}|}\sum_{\hat{p} \in \hat{P}} \min_{p \in P} \|\hat{p} - p\|_2\end{aligned} \tag{16}$$

which not requires that both P and $\hat{P}$ have the same number of points. $\min \|p - \hat{p}\|_2$ minimize the distance from point $p \in P$ to the closest point $\hat{p} \in \hat{P}$ and vice verse.

$$\mathcal{L}_{EMD}(P, \hat{P}) = \min_{\psi : P \to \hat{P}} \frac{1}{|P|}\sum_{p \in P} \|p - \psi(p)\|_2 \tag{17}$$

where $\psi : P \to \hat{P}$ is a bijection function to find the corresponding points using iterative $(1+\epsilon)$ approximation scheme. Therefore, $\mathcal{L}_{EMD}$ requires same number of points in P and $\hat{P}$.

Another one is that the pattern of output order is fixed, e.g., sorted, and the MSE loss $\mathcal{L}_{MSE}$ in (18) is performed on the corresponding index of the points.

$$\mathcal{L}_{MSE}(P, \hat{P}) = \frac{1}{|P|}\sum_{\substack{p_i \in P\\ \hat{p}_i \in \hat{P}}} \|p_i - \hat{p}_i\|_2 \tag{18}$$

where the same index i is representing the corresponding points in P and $\hat{P}$. Also, the same number of points is required. Thus, the reconstruction loss is applied as follows:

$$\begin{aligned}\mathcal{L}_{recons*} = \mathcal{L}_{MSE}(P, P_c) + \mathcal{L}_{EMD}(P, P_c)\\ + \gamma\mathcal{L}_{CD}(P, \hat{P})\end{aligned} \tag{19}$$

V. EXPERIMENTS

In this section, we first describe the implementation details of the encoder and decoder of our LPD-AE network, including parameters and function equations, which accomplish the latent space representation of the point cloud to achieve large-scale scene environment recognition and reconstruction tasks. Then, we give details of the training dataset and benchmark, including the public dataset, i.e., Oxford RobotCar [70] and KITTI datasets [71], and our freshly collected datasets. Finally, we show the performance of state of the art over existing methods and detailed ablation studies of our model.

A. IMPLEMENTATION DETAILS

As shown in Fig. 3, we give specific instances for recognition and reconstruction experiments with parameters and implementation functions for practical applications. The overall network structure is as described in Section IV.

For the encoder network in Section IV-B, the adaptive k ranging from [20, 100] to select the optimal neighborhood size and calculated structural features, and the dimensions are extended to $F = 64$dimensions through shared $MLP(64, 64)$. In the feature transformation, the equation $\boldsymbol{g}_\Theta$ is $MLP(64, 128, 1024)$, $\boldsymbol{l}_\Theta$ is $FC(512, 256, 64 * 64)$, and the feature transformation matrix A is obtained by the followed $Reshape$. Then in the feature aggregation, $k_\mathcal{F}$ and $k_\mathcal{C} = 20$, and the aggregation function $\boldsymbol{\phi}^e_\Theta$ is $MLP(64, 64)$. A $N \times 64$-dimensional matrix is obtained through the update function $\boldsymbol{\rho}^{e\rightarrow v}$: $Maxpooling$. Final a global description vector $f(P)$ of $\Gamma = 256$ dimensions is obtained through NetVLAD with $K_v = 64$ and $D_v = 256$.

For the decoder network in Section IV-C, rough key points are generated by the function $h_\Theta = FC(256, 256, N_c * 3)$ and mapping function ∇ : $Reshape$, and then the $Folding$ operation of $MLP(512, 512, 3)$ squeeze a mesh grid $\mathcal{M}(2, 0.05)$ to generate the dense point cloud $\hat{P} \subseteq \mathbb{R}^O$ in a $O \times 3$ matrix, $O = 4N_c$. In the $\mathcal{L}_{lazyQuad}$ of the recognition task, we set margins $\alpha = 0.5$, $\beta = 0.2$, $P_{pos} = 2$, $P_{neg} = 18$, and in the $\mathcal{L}_{recons*}$ of reconstruction task, to facilitate the $\mathcal{L}_{EMD}$ and $\mathcal{L}_{MSE}$ on coarse key points generation, we set $N_c = N$ and the weight γ of the $\mathcal{L}_{CD}$ is gradually increased from $\gamma = 0.01$ to 1 which utilized on dense points. The models are trained using Adam optimizer with an initial 0.0001, 0.7 times decayed every 50K steps learning rate. All experiments are conducted on a Titan XP GPU using TensorFlow.

B. DATASETS

We train and evaluate the LPD-AE on various datasets for scene recognition and reconstruction tasks, including extensive scenes, i.e., indoor, outdoor, campus, and urban environments.

1) PointNetVLAD BENCHMARK

We used the datasets and place recognition benchmark provided in PointNetVLAD. One of them, the outdoor large-scale scene dataset, Oxford RobotCar is used for training and validation of place recognition and reconstruction tasks. And place recognition evaluations are also performed on three other indoor datasets, a university sector (U.S.), a residential area (R.A.) and a business district (B.D.).

The Oxford RobotCar dataset is a whole 3D point cloud map built from 2D LIDAR and odometry and then cut into submaps, more details can find in PointNetVLAD [9]. Each submap is randomly downsampled to 4096 points with a voxel grid filter and normalized to [- 1,1]. It contains 44 data sets, which were collected on the same trajectory under different times, seasons, and weather conditions. Among them, 21711 submaps are used for training, and 3030 submaps are used for testing. The ground truth of place recognition is provided by corresponding UTM coordinates. The structurally similar and dissimilar point clouds for training in the recognition task, i.e., P_{pos} and P_{neg} are determined by the relative positions of two point clouds. The distance within 10 meters is defined as similar pairs, and the point clouds beyond 50 meters are most likely not similar.

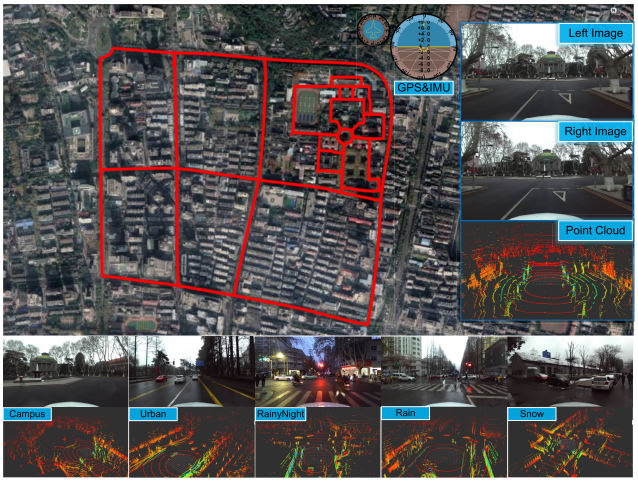

FIGURE 5. Demonstration of our dataset. We collect data on the same path of campus and city roads under different times and weather conditions, including binocular images, LIDAR point cloud, GPS, and IMU data, with a total length of nearly 50 kilometers.

2) KITTI DATASET

We performed place recognition validation and reconstruction training and validation on the KITTI dataset [71]. Unlike the Oxford RobotCar dataset, the KTTI dataset is urban environment data collected directly by Velodyne-64 LIDAR. According to the preprocessing method of Oxford RobotCar, we remove the ground, truncated it to 25 × 25 meters, and randomly downsampled to 4096 points with a voxel grid filter followed normalization. Only sequence 00 in odometry was used, 70% for training, and 30% for testing. However, KITTI is just collected on the same track at the same time.

3) OUR OWN DATASET

We design our own data acquisition platform to collect data using for long-term relocalization, place recognition, and reconstruction. And details can be found in Section VI-A.

Compared with the oxford RobotCar dataset, our own dataset has notable improvements on the acquisition method using a RoboSense-32 LIDAR and accurate ground truth. The GPS/INS fusion positioning system provides a decimeter-level position, and even post-processing can achieve centimeter-level accuracy. It also includes the

TABLE 1. Comparison results of the average recall (%) at top 1% and at top 1 (AR@1%/AR@1) under existing networks.

Dataset	LPD-AE	LPD-Net	PCAN (refine)	PCAN (baseline)	PN-VLAD (refine)	PN-VLAD (baseline)	PN-MAX	PN-STD
Oxford	93.35/83.78	**94.92 / 86.28**	86.32 / 70.85	83.81 / 69.05	80.71 / 63.33	81.01 / 62.76	73.87 / 54.16	46.52 / 31.87
U.S.	95.42/85.83	**96.00 / 87.04**	94.01 / 84.34	79.05 / 62.50	94.46 / 86.07	77.83 / 63.01	79.31 / 62.16	56.95 / 45.67
R.A.	90.19/81.73	90.46 / **83.06**	92.55 / 82.86	71.18 / 57.00	**93.07** / 82.66	69.76 / 56.19	75.14 / 60.21	59.81 / 44.29
B.D.	88.38/82.17	**89.14 / 82.31**	86.56 / 80.24	66.82 / 58.15	86.49 / 80.11	65.30 / 57.21	69.49 / 58.95	53.02 / 44.54

advantages of the Oxford RobotCar dataset, collecting scene data of different times and weathers on the same track of urban roads and campus as shown in Fig.5. The dataset also collects images of binocular cameras, GPS, IMU, and LIDAR data at the same time, which can facilitate a variety of research tasks, such as detection, relocalization, depth estimation, scene flow, 3D reconstruction, SLAM in dynamic scenes, etc.

We performed place recognition validation and reconstruction training and validation on our own dataset, which nearly 50 km route. We used 19927 point clouds, 70% for training, and 30% for testing as well.

C. PERFORMANCE

1) PLACE RECOGNITION

The evaluation of scene recognition performance uses the benchmark provided by PointNetVLAD. Specifically, in multiple sets of test data, we query the candidate point cloud with the closet distance in terms of the global descriptor. It regards as a successful place recognition if the point cloud retrieved within 25 meters between the ground truth. Therefore, the recall metric is used to evaluate the recognition accuracy of the model, with consistency to PointNetVLAD, we also use the Average Recall@N and Average Recall@1%. We compared with the existing work PointNetVLAD and PCAN, including the baseline and refine version. And also, compared with the original PointNet succeeded by maxpooling (PN-MAX) and the state of the art PointNet trained for classification on the ModelNet(PN-STD) to verify the generalization that the network trained on Modelnet can be extended to large-scale environments, which reported in [9].

The comparison results are shown in Fig.6 and Table.1. Our network, LPD-AE trained with multi-loss, significantly exceeds other comparison models and is also close to the original LPD-Net we proposed, with 93.4% recall at the top 1% on the Oxford RobotCar dataset. On the other three datasets, we still get better results without refinement than their refined model training with them. The 7% leap performance substantially confirms that our latent space representation extracts more efficient features and is generalizable to different point cloud datasets. The slight decrease in performance compared to LPD-Net is because the global descriptors generated by LPD-AE are trained not only for the place recognition but also for reconstruction. Furthermore, the practical application directly on KITTI and our dataset also verifies the feasibility of the model. For the comparison, PointNetVLAD and PCAN use the results of the pre-trained model provided by the author.

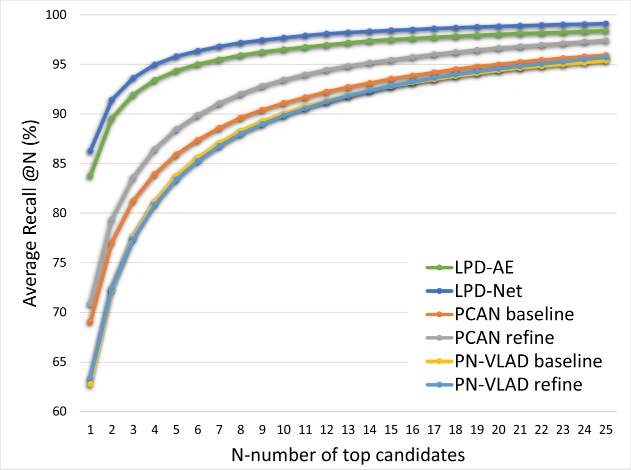

FIGURE 6. Average Recall of candidates comparison of existing works.

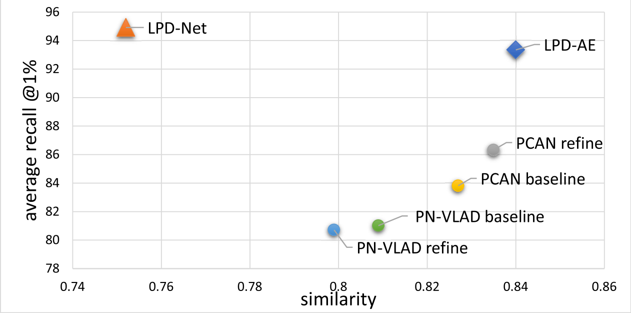

FIGURE 7. Average Recall and Similarity comparison of existing works.

In addition to comparing Average Recall, the Similarity of global descriptors is also considered to evaluate the performance, which calculated by the inner product of the descriptor vectors generated by the ground-truth recalls of the same place. A larger value of similarity indicates that the global descriptors corresponding to different point clouds in the same place are more similar or more stable. Average Recall and Similarity are a trade-off to some extent, and a more discriminative global descriptor is more conducive to improving the average recall. Therefore, the Average Recall of LPD-Net is quite high, but the Similarity is reversed, which reveals that the global descriptors of point clouds at the same place are easily confused. According to Fig.7 and Table.2, LPD-AE can balance both, improving the Similarity from 0.75 to 0.84 within the acceptable range of the Average Recall loss.

TABLE 2. Comparison results of the average recall (%) at top 1% (AR@1%), top 1 (AR@1) and Similarity under existing networks.

	AR@1%	AR@1	Similarity
PN-VLAD baseline	80.71	63.33	0.799
PN-VLAD refine	81.01	62.76	0.809
PCAN baseline	86.32	70.85	0.835
PCAN refine	83.81	69.05	0.827
LPD-Net	**94.92**	**86.28**	0.752
LPD-AE	93.35	83.78	**0.840**

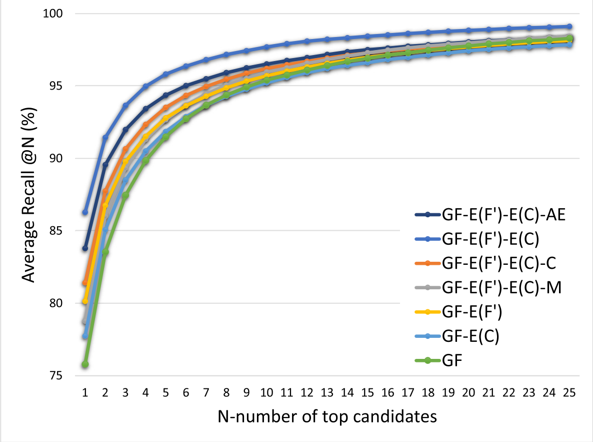

FIGURE 8. Ablation study results on different network structure.

TABLE 3. Ablation studies on different Network Structures.

	AR@1%	AR@1
$\boldsymbol{GF}$	89.77	75.79
$\boldsymbol{GF\_E_{(\mathcal{C})}}$	90.38	77.74
$\boldsymbol{GF\_E_{(\mathcal{F}')}}$	91.44	80.14
$\boldsymbol{GF\_E_{(\mathcal{F}')}\_E_{(\mathcal{C})}\_M}$	91.20	78.77
$\boldsymbol{GF\_E_{(\mathcal{F}')}\_E_{(\mathcal{C})}\_C}$	92.27	81.41
$\boldsymbol{GF\_E_{(\mathcal{F}')}\_E_{(\mathcal{C})}}$	**94.92**	**86.28**
$\boldsymbol{GF\_E_{(\mathcal{F}')}\_E_{(\mathcal{C})}\_AE}$	93.35	83.78

TABLE 4. Ablation studies on multi task and loss function.

	AR@1%	AR@1	Similarity
LPD-AE-recons-loss	76.32	61.95	**0.98**
LPD-AE-recog-loss	**94.92**	86.28	0.75
LPD-AE-multi-loss	93.35	83.78	0.84

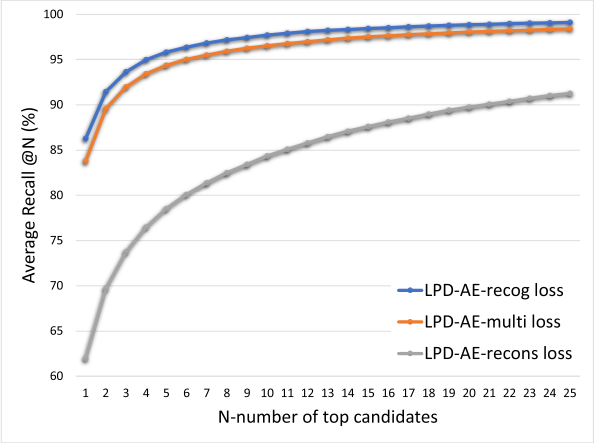

FIGURE 9. Ablation study results on Multi task and loss function.

a: DIFFERENT NETWORK STRUCTURE

We perform several removal and combination experiments on the Adaptive neighborhood geometric feature extraction($\boldsymbol{GF}$), Feature space aggregation $\boldsymbol{E}_{(\mathcal{F}')}$, and Physical space aggregation $\boldsymbol{E}_{(\mathcal{C})}$ in the network. The comparison results are shown in Fig.8 and Table. 3. $\boldsymbol{GF}$ shows the effectiveness of the adaptive neighborhood geometric feature, which is close to 9% improvement over PointNetVLAD. $\boldsymbol{E}_{(\mathcal{F}')}$ and $\boldsymbol{E}_{(\mathcal{C})}$ both contribute to the network, $\boldsymbol{C}$ represents the concatenate of parallel $\boldsymbol{E}_{(\mathcal{F}')}$ and $\boldsymbol{E}_{(\mathcal{C})}$ outputs, and $\boldsymbol{M}$ represents the maxpooling operation for parallel outputs. Still, the contribution of the combined $\boldsymbol{E}_{(\mathcal{F}')}\_\boldsymbol{E}_{(\mathcal{C})}$ is the most obvious, which proves our claim that characteristic features and their spatial distribution is sufficient representing large-scale scenes. $\boldsymbol{AE}$ represents for training using encoder networks and multiple loss functions as constraints, and the result shows that it also achieves fine performance in the place recognition task. Here we mainly focus on the contribution of $\boldsymbol{AE}$, more detailed analysis of network structures can be found in our previous work [12].

b: MULTI TASK AND LOSS FUNCTION

We study the effects of individual place recognition task, reconstruction task, and multi-task joint training on latent space representations. LPD-AE-recons-loss represents that the final output of the network is constrained only by reconstruction loss, and the global descriptor in the middle of the network has no constraints, while LPD-AE-recog-loss only applies the place recognition loss to constrain the global descriptor. The result in Table.4 and Fig.9 shows that the tendency of adopting reconstruction loss alone makes the descriptors more similar but reduces the distinguishability, and the recognition loss function alone is in contrast. Multi-task and multi-loss function training can guarantee both, indicating that the descriptors of latent space representation, used for place recognition and reconstruction, extract more reasonable scene features, rather than blindly improving the distinguishability.

c: DISCUSSION ON RECEPTIVE FIELD

The sufficient information for representative high-level feature extraction depends greatly on the corresponding receptive field, and the feature space and physical space aggregation methods we adopt can just expand the receptive field. As shown in Fig.4, the same local geometric features extracted by the neighborhood using the adaptive kNN can be clustered together in the feature space, and aggregated by the dynamic graph to enhance and extract more characteristic features. This is equivalent to the fact that parts with the

TABLE 5. FLOPs and model size comparison of existing networks.

	Parameters	FLOPs	Runtime per frame
PN-VLAD	19.78M	4.11G	13.09ms
PCAN	20.42M	23.01G	19.43ms
LPD-AE	19.81M	22.50G	24.57ms

same structure in physical space can be utilized to encode corresponding types of structural features. Then, the dynamic graph aggregation in physical space encodes the spatial distribution information of these structural features and also discover the knowledge of multiple structural features around the points to accumulate context features. Moreover, this also makes the receptive field of context features beyond the scope of a single structure or instance.

d: TIME AND SPACE COMPLEXITY ANALYSIS

We compare the required computational resources and model size with PointNetVLAD and PCAN in Table.5. We are comparable in size and computational resources to the PCAN model, but with much better performance. The real-time nature of the model can satisfy the actual lidar application.

2) RECONSTRUCTION

The reconstruction task was trained and validated on Oxford RobotCar, KITTI, and our own dataset, using the global vector generated by the encoder, LPD-net. However, there is no relevant benchmark for large-scale point cloud reconstruction, especially for recovery from global descriptors. We use Chamfer Distance and Earth Mover's Distance as evaluation metrics on the test dataset.

We give a qualitative analysis through the visual comparison of the reconstructed point cloud and the original point cloud. And a quantitative comparison with different encoder networks and reconstruction settings. The decoder network can recover the original point cloud contour shape through a compact global descriptor, and it shows that the latent space representation truly discovers the characteristics of the large-scale environment.

a: QUALITATIVE VISUALIZATION

We demonstrate that the point cloud of the large-scale scene can be recovered from the global descriptor with a length of 256, and the compression ratio can be 48 : 1. It is unmanageable for us to give a quantitive comparison because very little work has been done on the compression and reconstruction of large scenes. We evaluate LPD-AE on multiple datasets and show some representative visualizations of reconstructed point clouds and corresponding distance metric. On the Oxford RobotCar test dataset, LPD-AE produces the dense point cloud with 16384 points rather than 4096 points, and therefore evaluated it by the Chamfer Distance, with the overall result is $CD = 0.018$. From the visualization in Fig.10, our LPD-AE network can recover the main contour shape of the point cloud of the environment, which verifies that our encoder network has learned the features for latent space representation of point clouds. These features are very intuitive and difficult to characterize with specific equations. Nevertheless, it can be perceived that the large-scale point cloud compression and reconstruction task is particularly challenging considering the reconstructed point cloud is more blurred than the original point cloud and lacks structural details.

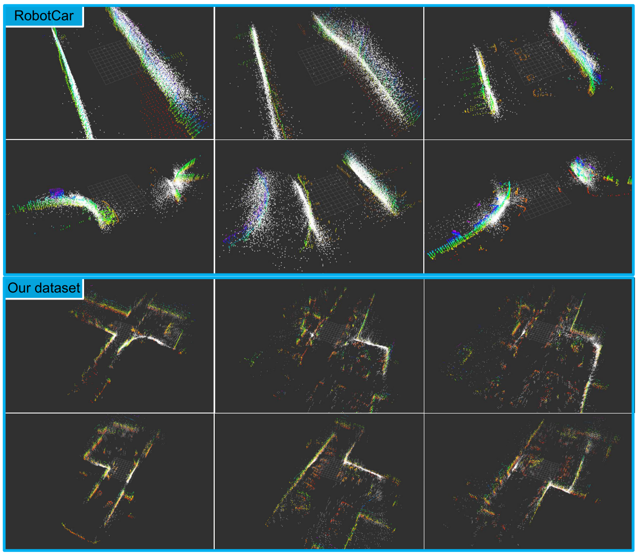

FIGURE 10. Qualitative visualization for reconstruction results. It can be seen that the reconstruction restores the basic outline shape of the point cloud, but it is relatively rough.

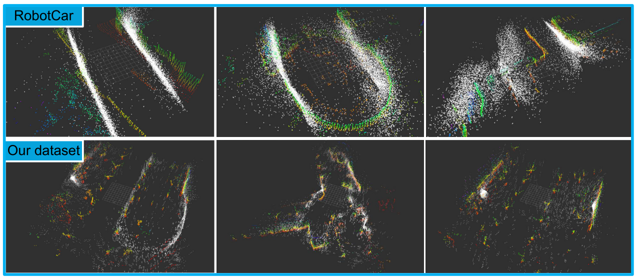

FIGURE 11. Failure cases of Our LPD-AE reconstruction. The point clouds with relatively large reconstruction errors are basically small samples.

We observe that the network pays more attention to significant components in the environment, such as buildings, and is not sensitive to small objects, such as people and cars. The results also confirm the robustness of the features LPD-AE extracted from dynamic scenes.

We also give some failure cases in Fig.11, which may be due to the uneven sample of the dataset, fewer samples at the intersection, and larger reconstruction errors on the corner structure.

b: DIFFERENT KEYPOINTS NUMBERS

We study the effect of the number of rough keypoints on reconstruction accuracy. Comparisons in Fig.12 and Table.6

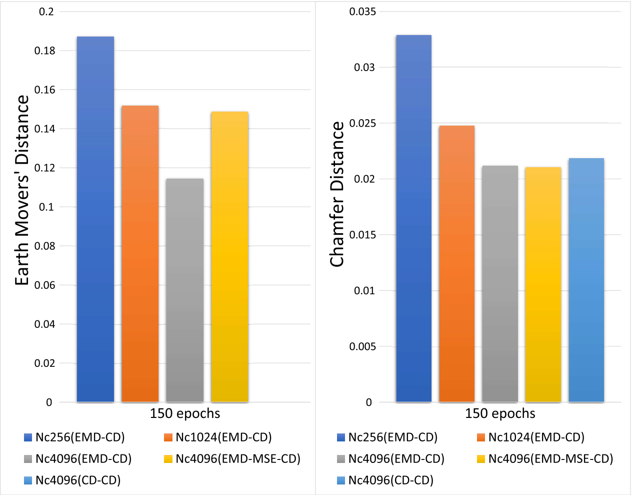

FIGURE 12. Quantitative comparison results on different numbers of key points. For both EMD and CD metric, lower is better. (EMD-MSE-CD) means training with $\mathcal{L}_{EMD} + \mathcal{L}_{MSE} + \mathcal{L}_{CD}$.

TABLE 6. Quantitative comparison results on different keypoints numbers.

	EMD	CD
$N_c = 256$ $(\mathcal{L}_{EMD}+\mathcal{L}_{CD})$	0.187	0.0329
$N_c = 1024$ $(\mathcal{L}_{EMD}+\mathcal{L}_{CD})$	0.152	0.0247
$N_c = 4096$ $(\mathcal{L}_{EMD}+\mathcal{L}_{CD})$	**0.114**	0.0212
$N_c = 4096$ $(\mathcal{L}_{EMD}+\mathcal{L}_{MSE}+\mathcal{L}_{CD})$	0.149	**0.0210**
$N_c = 4096$ $(\mathcal{L}_{CD}+\mathcal{L}_{CD})$		0.0218

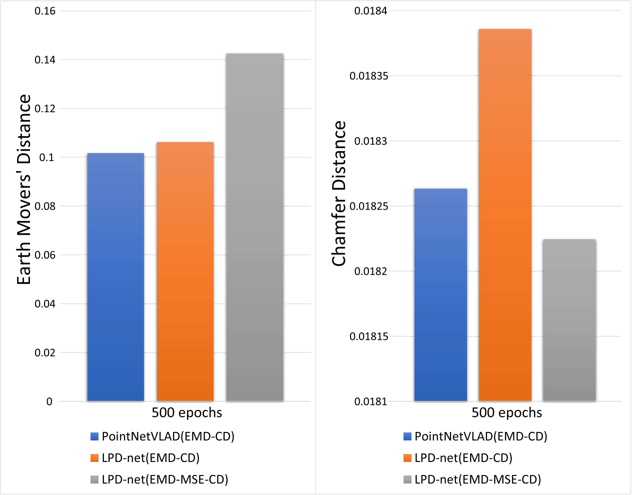

FIGURE 13. Quantitative comparison results on different Loss function and Encoder network. Also, lower is better.

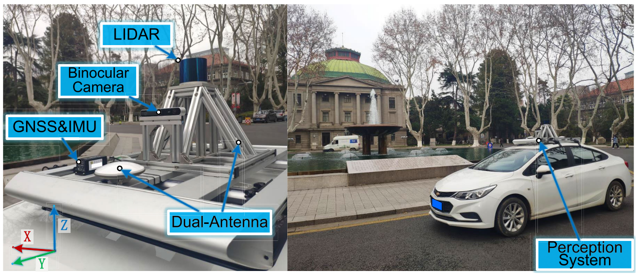

FIGURE 14. Multi-sensor fusion hardware platform for data collection and application.

show that the more rough-keypoints, the higher the accuracy of the reconstructed dense point cloud. All experiments use $\mathcal{L}_{EMD}$ for rough points, $\mathcal{L}_{CD}$ for dense points, and 150 epochs of training for fair comparisons.

c: DIFFERENT LOSS FUNCTION

We validate the effectiveness of different loss function combinations in the reconstruction task. Because the number of generated points O exceeds the number of ground truth N, the $\mathcal{L}_{CD}$ is the only option for the dense point cloud generation, and for the key points generated in the intermediate stage, $\mathcal{L}_{CD}$, $\mathcal{L}_{EMD}$, and $\mathcal{L}_{EMD}$ can be the alternative. Given the fact that the combination of $\mathcal{L}_{EMD}$ and $\mathcal{L}_{EMD}$ is the best result of reconstruction accuracy from Fig.13, it can be inferred that these two loss functions constrain the key points in space location and spatial distribution, which can facilitate reconstruction of dense point clouds.

d: PointNetVLAD VS LPD-Net

We replaced the encoder of the auto-encoder architecture with PointNetVLAD to study the effect of different encoders on the ability of reconstruction with the corresponding intermediate codeword. Compared with PointNetVLAD, the reconstruction evaluation metric is comparable, or PointNetVLAD is slightly better in Fig.13. Still, our network is superior to PointNetVLAD in the experiment of the map compression and reconstruction as depicted in Fig.17, since the LPD-AE extracts more reasonable environmental features.

VI. APPLICATIONS

A. LATENT SPACE ANALYSIS ON OUR OWN PLATFORM

1) PLATFORM

We develop a multi-sensor fusion hardware system to collect data and practical applications, as disclosed in Fig.14, which equipped with RoboSense-32 LIDAR, ZED binocular camera with 120mm baseline, InertialLabs INS-D dual-antenna GPS / INS integrated positioning and navigation system, etc. Through hardware synchronization and system calibration, it can provide point clouds, dual-channel images, raw GPS and IMU signals, fusion-corrected position and attitude signals, and fusion odometer signals. We will release the dataset created by the system with ground truth position and pose as described in Section V-B3 for a variety of tasks.

The system is also equipped with an industrial computer with a GPU, which provides the feasibility for the research on autonomous driving perception system. We carry out the

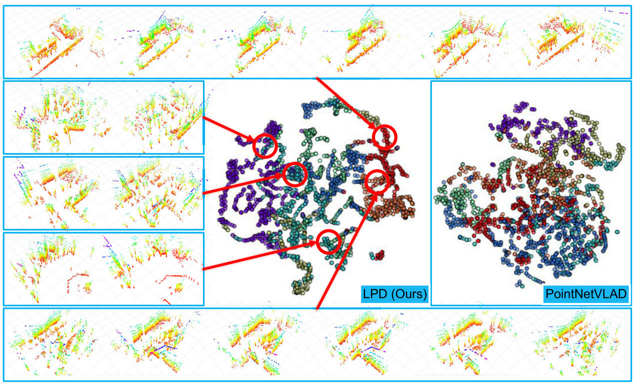

FIGURE 15. t-SNE visualization of latent space. Compared with PointNetVLAD, the latent space we generate is more distinguishable. The point clouds in the blue box are examples of clustering, which indicate that the point clouds with similar structures are closer in the latent space.

practical application of SLAM on this platform to confirm the validity and practicability of our proposed latent space representation. The followed experiments show the powerful potential of our model in practical applications.

2) LATENT SPACE VISUALIZATION

We exhibit the visualization of latent space using t-SNE on our own datasets. From the visual comparison of latent space in Fig.15, it can be demonstrated that the descriptors generated by LPD-AE are more distinguishable than PointNetVLAD, and the distance between similar point clouds closer, dissimilar point clouds have clear boundaries. The latent space representation of point clouds in large-scale scenes provides reliable and quantifiable metrics, which can determine whether they are structurally similar or not through the L2 distance of the descriptor from two frames of point clouds. Examples of clustered point cloud corresponding to latent space depict the distribution of point clouds in our dataset, and it can be revealed that similar point clouds are more compact in the latent space.

3) LOOP CLOSURE DETECTION AND RELOCALIZATION VIA PLACE RECOGNITION

With the benefit of our platform, we perform live loop closure detection and relocalization applications through place recognition. The overall system of the experiment is shown in Fig.2, and details described in Section III. During the SLAM mapping process, the keyframes that have been reached are retrieved through the global descriptor to detect the loop closure. Accordingly, the $L2$ distance matrix of the keyframes descriptors and the map modified using the loop closure are shown in Fig.16, and the blue keyframes represent the corresponding loop closure paths detected, like [72]. For relocalization, we utilize point clouds of the same trajectory at different times to retrieve in the built maps, with corresponding images and point clouds presented in Fig.16. The positioning capabilities based on global features can also be extended to multi-robot joint mapping similar to [18], locating itself in other's maps with the assistance of the global descriptor.

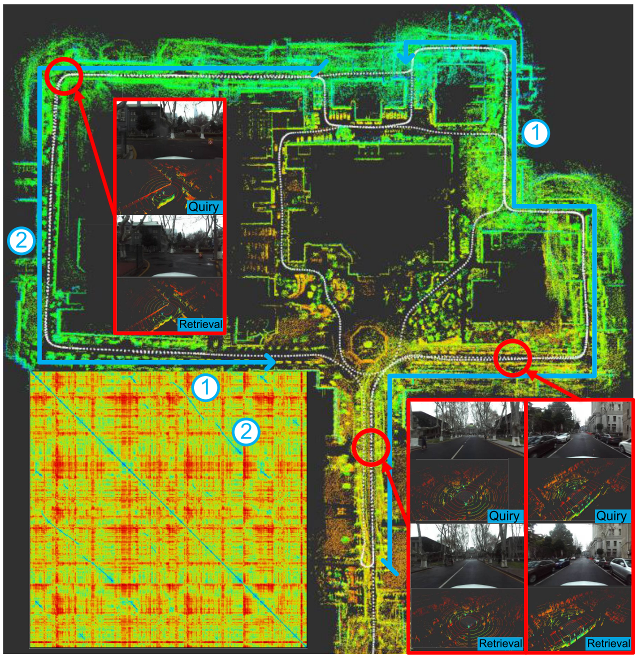

FIGURE 16. Results of loop closure detection and relocalization. The loop closure detection in the $L2$ distance matrix figure (the lower left) corresponds to the blue path. The red box is the result of relocalization based on the point cloud at different times, and the image is only for display reference.

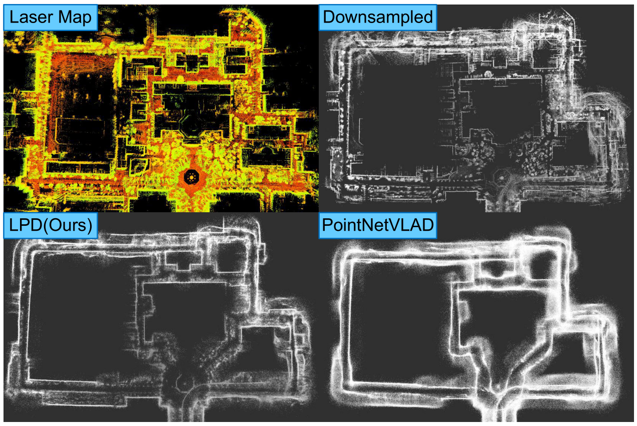

FIGURE 17. Visual comparison of reconstruction results. The upper left figure is the SLAM map of our dataset, and the upper right figure is the keyframe map of the pre-processed and down-sampled point clouds. The bottom left corner is the keyframe map reconstructed by our LPD model as the encoder, and the lower right corner is the keyframe map reconstructed using PointNetVLAD.

4) COMPRESSING STORAGE AND RECONSTRUCTION OF LARGE-SCALE POINT CLOUD MAP

Fig.17 manifests the compression and reconstruction capabilities of the LPD-AE. We compress and reconstruct the point cloud of each keyframe of the map built in the experiment above, and rebuild the map using the reconstructed keyframes and corresponding position and attitude information. Impressively, LPD-AE demonstrates the remarkable aptitude to reconstruct dense point clouds from such compressed features.

VII. CONCLUSION

In this paper, we present a novel and practical pipeline of latent space representation for large-scale point clouds. With the capabilities of the proposed LPD-AE network, the irregular point cloud can be converted to equivalent latent spaces and also reconstructed. Simple descriptor calculations are used to achieve tasks that are initially computationally complex, such as place recognition, loop closure detection, relocalization, and compressed transmission, with less memory, computing resources, and transmission bandwidth.

What's more, comprehensive evaluations and extensive applications have demonstrated impressive and remarkable performance to reach the state of the art, which manifests the great potential for applications in robotic and autonomous driving.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probalistic Robotics*. Reading, U.K.: Kybernetes, 2006.

[2] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, "Large-scale location recognition and the geometric burstiness problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1582–1590.

[3] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[4] M. J. Cummins and P. M. Newman, "Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 3–10.

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NIPS*, 2018, pp. 5105–5114.

[7] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," 2018, *arXiv:1801.07829*. [Online]. Available: http://arxiv.org/abs/1801.07829

[8] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

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

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

[11] W. Zhang and C. Xiao, "PCAN: 3D attention map learning using contextual information for point cloud based retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12436–12445.

[12] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y. Liu, "LPD-net: 3D point cloud learning for large-scale place recognition and environment analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2831–2840.

[13] N. Snavely, "Scene reconstruction and visualization from Internet photo collections: A survey," *IPSJ Trans. Comput. Vis. Appl.*, vol. 3, pp. 44–66, Dec. 2011.

[14] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A survey of urban reconstruction," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 146–177, 2013.

[15] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 605–613.

[16] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 206–215.

[17] J. Li, B. M. Chen, and G. H. Lee, "SO-net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[18] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D segment mapping using data-driven descriptors," 2018, *arXiv:1804.09557*. [Online]. Available: http://arxiv.org/abs/1804.09557

[19] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Aug. 2011, pp. 2564–2571.

[21] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proc. IEEE Eur. Conf. Comput. Vis. (ICCV)*, May 2006, pp. 404–417.

[22] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[25] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[26] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*. Prague, Czech Republic: Springer, 2004, pp. 224–237.

[27] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3384–3391.

[28] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.

[29] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique signatures of histograms for surface and texture description," *Comput. Vis. Image Understand.*, vol. 125, pp. 251–264, Aug. 2014.

[30] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[31] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep panoramic representation for 3-D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.

[32] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.

[33] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.

[34] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.

[35] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2088–2096.

[36] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Point CNN: Convolution on x-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 820–830.

[37] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.

[38] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. II–3, pp. 181–188, Aug. 2014.

[39] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1808–1817.

[40] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4830–4836.

[41] R. Dube, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5266–5272.

[42] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[43] Z. Sivic, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[44] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1578–1585.

[45] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[46] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[47] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3384–3391.

[48] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Feb. 2012, pp. 2911–2918.

[49] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[50] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[51] M. Cummins, "Highly scalable appearance-only Slam-fab-map 2.0," in *Proc. Robot., Sci. Syst. (RSS)*, 2009, pp. 1–15.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[53] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot., Sci. Syst.*, Jun. 2009, p. 435.

[54] Z. Liu, C. Suo, S. Zhou, F. Xu, H. Wei, W. Chen, H. Wang, X. Liang, and Y.-H. Liu, "SeqLPD: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1218–1223.

[55] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning the matching of local 3D geometry in range scans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2017, pp. 199–208.

[56] R. Huang, P. Achlioptas, L. Guibas, and M. Ovsjanikov, "Latent space representation for shape analysis and learning," 2018, *arXiv:1806.03967*. [Online]. Available: http://arxiv.org/abs/1806.03967

[57] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," 2017, *arXiv:1707.02392*. [Online]. Available: http://arxiv.org/abs/1707.02392

[58] M. Zamorski, M. Ziäba, P. Klukowski, R. Nowak, K. Kurach, W. Stokowiec, and T. Trzciáski, "Adversarial autoencoders for compact representations of 3D point clouds," *Comput. Vis. Image Understand.*, vol. 193, Apr. 2020, Art. no. 102921.

[59] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[60] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2018, pp. 728–737.

[61] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-net: Point cloud upsampling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2790–2799.

[62] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "PU-GAN: A point cloud upsampling adversarial network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7203–7212.

[63] R. Hartley and A. Zisserman, *Multiple View Geometry Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[64] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[65] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4631–4640.

[66] L. Yi, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, p. 210, Nov. 2016.

[67] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2005, pp. 539–546.

[68] M. Jaderberg, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[69] P. W. Battaglia, "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*. [Online]. Available: http://arxiv.org/abs/1806.01261

[70] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, Jan. 2017.

[71] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[72] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1643–1649.