Fast and Accurate Deep Loop Closing and Relocalization for Reliable LiDAR SLAM

Chenghao Shi, Xieyuanli Chen, Junhao Xiao, *Senior Member, IEEE*, Bin Dai, and Huimin Lu

***Abstract*—Loop closing and relocalization are crucial techniques to establish reliable and robust long-term SLAM by addressing pose estimation drift and degeneration. This article begins by formulating loop closing and relocalization within a unified framework. Then, we propose a novel multihead network, LCR-Net, to tackle both tasks effectively. It exploits novel feature extraction and a pose-aware attention mechanism to precisely estimate similarities and 6-DoF poses between pairs of LiDAR scans. In the end, we integrate our LCR-Net into a SLAM system and achieve robust and accurate online LiDAR SLAM in outdoor driving environments. We thoroughly evaluate our LCR-Net through three setups derived from loop closing and relocalization, including candidate retrieval, closed-loop point cloud registration, and continuous relocalization using multiple datasets. The results demonstrate that LCR-Net excels in all three tasks, surpassing the state-of-the-art methods and exhibiting a remarkable generalization ability. Notably, our LCR-Net outperforms baseline methods without using a time-consuming robust pose estimator, rendering it suitable for online SLAM applications. To the best of the authors' knowledge, the integration of LCR-Net yields the first LiDAR SLAM with the capability of deep loop closing and relocalization.**

***Index Terms*—3-D registration, autonomous driving, deep learning, loop closing, relocalization.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM), plays a fundamental role across domains such as autonomous driving, robotics, and computer vision. Ensuring the reliability and stability of a SLAM system is crucial for practical applications. External sensors such as global positioning system (GPS) and inertial measurement unit (IMU) are commonly employed to enhance SLAM in real-world scenarios, but challenges arise when they are inaccessible or unreliable. Hence, attaining enduring reliability in LiDAR-only SLAM becomes important yet challenging. Relocalization and loop closing are pivotal techniques within this context. Relocalization refers to recovering the relative 6 degrees of freedom (DoF) transformation between the current sensor frame to the global coordinate system (previously generated map), especially when local tracking failures, whereas loop closing involves identifying previously visited locations to correct the drift in pose estimation. Despite different objectives, both techniques share similar underlying concepts. They both first coarsely locate the most similar candidate in the map relative to the current scan, followed by a fine relative pose estimation. More specifically, existing methods [1], [2], [3] achieve full 6-DoF LiDAR loop closing using bag-of-words (BoW) [4]. Such methods extract local features to construct BoW models for loop closure detection. The poses are then obtained by matching the local features and then registering.

Since deep neural networks have shown great advances in perception tasks, recent studies have also employed deep learning approaches [5], [6], [7] for LiDAR loop closing. They follow a similar structure as BoW, albeit using learning-based approaches instead of hand-crafted local features extraction and BoW model. These techniques face a dilemma: obtaining a comprehensive representation of environmental features often requires a deeper encoder. However, a deeper encoder diminishes the number of local features, potentially hindering accurate localization. In addition, improving registration performance often involves integrating more complex designs into local feature extraction, thereby, substantially reducing global description efficiency. This dilemma is particularly critical in relocalization, where accurate localization and rapid global description are needed. Therefore, despite loop closing and relocalization sharing similar underlying techniques, few works have specifically been proposed for LiDAR relocalization. To the best of the authors' knowledge, no prior research has addressed both LiDAR loop closing and relocalization simultaneously. In this article, we introduce LCR-Net, a novel multihead network tackling both LiDAR loop closing and relocalization using a joint framework (as in Fig. 1), offering four main contributions:

First, we revisit the challenges of addressing LiDAR loop closing and relocalization separately. We identify limitations within the current paradigm and propose *a new framework for solving loop closing and relocalization simultaneously* (Section III). Our framework leverages the shared techniques

Manuscript received 4 February 2024; accepted 22 March 2024. Date of publication 8 April 2024; date of current version 29 April 2024. This paper was recommended for publication by Associate Editor A. Nuechter and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB4701600, in part by the National Science Foundation of China under Grant U1913202, Grant U22A2059, and Grant 62203460, in part by the Fund for key Laboratory of Space Flight Dynamics Technology under Grant 2022-JYAPAF-F1028, in part by Young Elite Scientists Sponsorship Program by CAST under Grant 2023QNRC001, and in part by the Major Project of Natural Science Foundation of Hunan Province under Grant 2021JC0004. *(Chenghao Shi and Xieyuanli Chen contributed equally to this work.) (Corresponding authors: Junhao Xiao; Huimin Lu.)*

Chenghao Shi, Xieyuanli Chen, Junhao Xiao, and Huimin Lu are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410003, China (e-mail: junhao.xiao@nudt.edu.cn; lhmnew@nudt.edu.cn).

Bin Dai is with the National Innovation Institution of Defense Technology, Beijing 100073, China.

Data is available online at: https://github.com/nubot-nudt/LCR-Net.

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2024.3386363, provided by the authors.

Digital Object Identifier 10.1109/TRO.2024.3386363

1941-0468 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

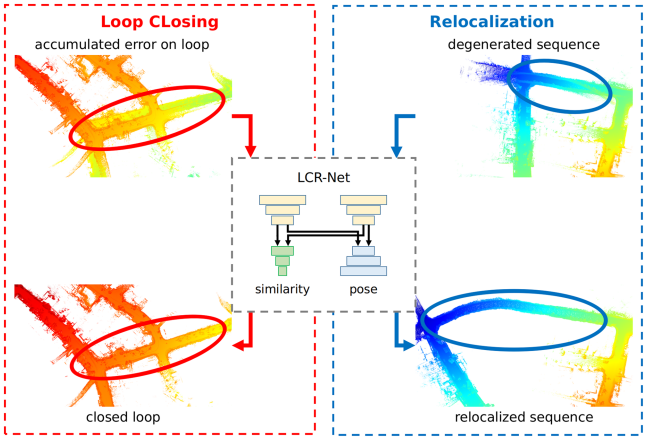

Fig. 1. Our proposed LCR-Net for loop closing and relocalization. LCR-Net solves both tasks by first retrieving the most similar frame from the map and then estimating the 6-DoF pose.

underlying these two tasks, integrating them within a coherent coarse-to-fine framework. This framework concurrently addresses both tasks, beginning with generating global descriptors for an initial coarse global candidate search. Subsequently, the framework generates dense local features to facilitate precise 6-DoF pose estimation. Through such a framework, we circumvent dilemmas arising from the requirements of these two tasks, providing a solid foundation for improved candidate retrieval and registration performance.

Following our framework, we introduce *a novel multihead network LCR-Net* (Section IV). It employs a shared backbone to encode the point cloud into three types of sparse features. These features are then processed separately in two distinct heads. One head generates a lightweight global descriptor for each scan, enabling fast candidate retrieving. The other head establishes dense matches in a sparse-to-dense manner for accurate registration. Unlike existing methods using only sparse features, our approach exploits dense feature matching based on the neighborhood consistency, achieving accurate and fast pose estimation for online applications without requiring costly robust estimator or iterative pose refinements. To effectively train the multihead network, we also introduce novel losses with a specific training strategy, which leads to state-of-the-art (SOTA) performance surpassing all existing baselines.

To enhance the performance on both tasks, we present the third contribution as the *novel keypoint detection module comprising two novel submodules, 3D-RoFormer++ and VoteEncoder* (Section IV-A). The keypoint detection module offers three types of features, enabling fast global description and reliable dense matching. Initially, it rapidly samples and encodes uniform features across the point cloud for overall representation of the environment. Subsequently, to enhance the features and make them salient and discriminative for improving registration, we introduce 3D-RoFormer++ and VoteEncoder. 3D-RoFormer++ enhances the representation capability of local features with contextual and structural information by enabling the information exchange between the two point clouds. The other module, VoteEncoder effectively downsamples the points while identifying keypoints lying on geometrically significant regions and subsequently aggregating the features from their neighbors. The VoteEncoder can significantly enhance registration robustness and accuracy by improving the coverage of matching points over the overlapping area of point clouds. The superior improvement brought by VoteEncoder highlights the importance of the match distribution OVER inlier ratio (IR), offering valuable insights (Section VI-H) for future research on point cloud registration.

The fourth contribution is *a novel LiDAR SLAM system with the capability of deep learning-based loop closing and relocalization* (Section V). We build a full LiDAR SLAM system based on our proposed deep loop closing and relocalization method. The system effectively tackles the local pose tracking, loop closing, and relocalization in parallel. A thorough evaluation of our SLAM system in diverse environments showcases the effectiveness and robustness after integrating LCR-Net.

We extensively evaluate our approach on three setups derived from loop closing and relocalization: candidate retrieval, closed-loop point cloud registration, and continuous relocalization. The results demonstrate the following.

1) Our approach outperforms respective baselines and dominates the SOTA in all three tasks, in particular;
2) our approach achieves the best candidate retrieval performance with a simple architecture benefiting from the feature representation capability of the backbone;
3) our approach boosts the baseline by a large margin in registration tasks, even outperforms the baseline method refined by iterative closest point (ICP) [8]; and
4) the SOTA registration performance can be achieved efficiently without requiring for a costly random sample and consensus (RANSAC) estimator.

We also conduct tests on multiple sequences to assess the performance of our SLAM system integrating LCR-Net. The results depict that our SLAM system is capable of addressing the relocalization and loop closing challenges. In the loop closing task, our approach outperforms the most commonly employed loop closing approach, Scan Context [9] combined with ICP. In addition, we provide detailed ablation studies to demonstrate the effectiveness of our design.

II. Related Work

While various studies have been conducted for the foundational techniques underlying loop closing and relocalization, including candidate retrieval and point cloud registration, few works can simultaneously address both tasks. Therefore, we first introduce the underlying techniques and then explore the recent advancements in loop closing and relocalization.

Candidate retrieval, also known as place recognition or loop closure detection, compares the current sensor observation with prebuilt maps to determine the approximate location of the robot within the map. LiDAR-based loop closure detection approach can be categorized into local descriptor-based methods and global descriptor-based methods.

Local feature-based methods typically extract local sparse features from point clouds [10], [11] and organize them using

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

a BoW model for place recognition [2], [3]. These methods are capable for 6-DoF pose estimation based on local feature correspondences. However, extracting stable and reliable local features from 3-D LiDAR scan is a challenging task, which limits the performance of such methods. The first work employing deep learning for point cloud retrieval, PointNetVLAD [12], extracts local features using PointNet [13] and then aggregates them into global descriptors using NetVLAD [14]. There are also methods [15], [16], [17], which exploit sparse 3-D convolution for local feature extraction, but use different pooling strategy for global descriptor generation, i.e., generalized-mean pooling [18] for MinkLoc3D [15], [16] and second-order pooling for LoGG3D-Net. Transformer [19] is also explored in generating global descriptors [20], [21]. Recently, LCDNet [5] employed PVRCNN [22] for robust feature extraction and then utilized NetVLAD for global descriptor generation. FinderNet [23] circumvents the challenge of feature extraction from point clouds by converting them into digital elevation maps (DEMs) and leveraging a CNN network to extract local features and global descriptors. However, converting point clouds to DEM makes it challenging to estimate 6-DoF poses. In contrast, our method operates directly on point clouds, bypassing the challenge of estimating pose on sparse local features while possessing both loop closure detection and accurate 6-DoF pose estimation capability.

Global descriptor-based methods, such as Scan Context [9], represent point clouds as overhead views and encode different segmented spaces to construct global descriptors. Wang et al. [24] extracted descriptors using LoG-Gabor threshold filtering and measure similarity using Hamming distance. OverlapNet [25] introduces a deep learning-based method, which estimates the overlap and relative yaw angles of a set of point clouds for place recognition and initial pose estimation. Ma et al. [26], [27], [28] combined OverlapNet with Transformer to propose a rotation-invariant global descriptor. While the global descriptor-based methods can identify loop closures, they lack the ability to estimate the 6-DoF pose between the current scan and the loop candidate.

Point cloud registration refers to determine the relative spatial transformation that aligns two point clouds. Extracting accurate correspondence is the most challenging aspect. Once correspondences are established, the transformation can be solved using either a direct solver or a robust estimator [29]. The ICP algorithm [8] and its various variants [30], [31] are known and applied methods. These methods establish correspondences iteratively using nearest neighbor search or other heuristics. However, the common drawback among ICP-like methods is that they heavily rely on good initial estimates for the transformation.

To release the requirement of initial estimates, other methods opt to establish correspondences on local features [11], [32] to achieve global registration. Due to the powerful feature representation capabilities exhibited by deep learning, massive learning-based methods for feature extraction have been proposed. Deng et al. [33], [34] proposed PPFNet and PPF-FoldNet, which combine point pair features (PPF) with PointNet [13] to generate local patch representations for matching. In contrast to PPFNet and PPF-FoldNet, which establish correspondences on uniformly sampled points, keypoint-based techniques sample points based on predefined [11] or learned saliency [35], [36], [37], [38] to achieve better repeatability. Due to the inherent errors introduced by individual matches, establishing matches on sparse keypoints generated either through uniform sampling or keypoint detection can limit registration accuracy. Recently, some studies [39], [40], [41] employed a coarse-to-fine mechanism that initially seeks correspondences on sparse keypoints and then extends them to dense ones, showing potential in registration. To enhance the reliability of sparse keypoint correspondences, CoFiNet [39] exploits Transformer for contextual information aggregation and GeoTransformer [40] introduces geometric transformer to incorporate relative geometric information. RDMNet [41] introduces 3D-RoFormer for fast and lightweight relative geometric information encoding and the voting scheme for keypoint detection. HRegNet [42] extracts multilevel features and refines the transformation hierarchically. Despite the rapid advancements in global registration methods, these studies have remained disconnected from the challenges of loop closing and relocalization, lacking the ability of similarity evaluation.

Loop Closing and Relocalization both need to first find a coarse location and then estimate the fine 6-DoF pose. Though the individual techniques are widely explored, as discussed in the previous review, few works can address both tasks simultaneously. OverlapNet [43] and FinderNet [23] are capable of estimating 1-DoF or 3-DoF poses while implementing loop closure detection. These methods have also shown success in achieving 6-DoF loop correction when combined with other local registration techniques. However, in more challenging scenarios involving large pose drift or relocalization, non-6-DoF pose estimation is insufficient. There are both traditional [3] and learning-based [5], [44], [45] methods that achieve loop closure detection and 6-DoF pose estimation by extracting local features. Nevertheless, the accuracy of sparse local feature-based registration is constrained, requiring additional refinement through local registration techniques such as ICP. Therefore, these methods are evidently unsuitable for relocalization tasks where local registration has already failed and rapid online pose recovery is required.

To address system degeneration, most methods integrate additional sensors such as camera [46], IMU [47], or ultrawideband [48], and switch between different tracking modalities, whereas some methods [49], [50] fall back from frame-to-map to frame-to-frame pose estimation to avoid errors caused by distorted maps. However, to the best of the authors' knowledge, no piror LiDAR-only method has been proposed to achieve relocalization handling system degeneracy. This can be attributed to the challenge in achieving accurate LiDAR-based global registration when local pose tracking has already failed.

In the field of visual SLAM, however, due to the success in visual feature extraction, loop closing and relocalization have been well-explored. Oriented fast and rotated BRIEF (ORB)-SLAM [51] extracts ORB features to form a BoW model for loop closing. When local tracking fails, ORB-SLAM switches from frame-to-frame alignment to frame-to-map alignment to

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

find more potential landmark matches for relocalization. Object aided (OA)-SLAM [52] achieves more robust relocalization by relocalizing with reconstructed objects instead of local landmarks. Our approach, however, seeks to generate global descriptors to achieve rapid similarity evaluation for candidate detection in loop closing and relocalization. In addition, a robust and accurate enough global registration method is employed to achieve 6-DoF pose estimation.

III. Problem Definition

We aim to address the challenges of loop closing and relocalization for LiDAR-based SLAM in outdoor driving environments. The underlying techniques of relocalization and loop closing are similar: both tasks involve the identification of the most similar candidate scan from the existing map and subsequently determining the 6-DoF pose. This commonality provides a foundation for addressing both tasks within a unified framework. However, the technical focus of the two tasks is quite different. First, in loop closing, the main challenge lies in rapidly and accurately identifying loop closures within a large database. In most cases, loop closures are identified when there is a substantial overlap between the current and candidate scans, simplifying the subsequent registration process once the loop closure has been correctly identified. Conversely, selecting the candidate scan for relocalization is relatively straightforward. In many autonomous driving situations, simply opting for the most recent scans can be sufficient. Instead, the primary challenge of relocalization lies in achieving precise and robust global registration, as local pose tracking, even with prior information, has failed in such situations. This typically occurs in challenging scenarios that involve low overlap, extensive occlusions, or degraded scene features, posing significant challenges for point cloud registration. Second, loop closing can be executed at a relatively low frequency as a few correctly closed loops are sufficient to eliminate accumulated error. However, relocalization needs to be fast as it directly affects online localization. A prolonged relocalization process reduces the overlap between the current scan and the map, diminishing the success rate. While numerous works have focused on loop closing, the demanding requirements of robustness, accuracy, and speed in registration explain the limited attention given to relocalization. To address this issue, we aim to initially study the framework to support the requirements of both tasks.

A commonly employed framework for simultaneously similarity evaluation and registration is shown in Fig. 2(a). For an incoming LiDAR scan $\mathcal{P} = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^{N}$, it initially downsamples and encodes the point cloud into local features $[\hat{\mathcal{P}}|\hat{\mathbf{F}}] = f_{\text{backbone}}(\mathcal{P})$, and then generate a global descriptor $\boldsymbol{V} = f_{\text{Encoder}}(\hat{\mathcal{P}}, \hat{\mathbf{F}})$ based on these local features. The global descriptor $\boldsymbol{V}$ is exploited to exhibit similarity for candidate frame retrieving and the local features $[\hat{\mathcal{P}}|\hat{\mathbf{F}}]$ are matched for pose estimation. Such a framework faces a dilemma: A deeper backbone is often needed to ensure reliable global feature representations. However, this can result in a reduced number of local features, which could harm registration performance. On the other hand, incorporating more complex designs into the backbone to enhance registration performance can often result in reduced global description efficiency. This is a crucial consideration in candidate retrieval tasks. Based on this insight, we propose the framework shown in Fig. 2(b). We leave the workflow of global descriptor generation untouched but incorporate a decoder with skip connections for denser local feature generation $\mathbf{F} = f_{\text{Decoder}}(\hat{\mathcal{P}}|\hat{\mathbf{F}})$. Introducing a decoder for feature generation is not new technically. Existing methods [17], [45] employ an encoder/decoder structure in the backbone for extracting local features. However, these frameworks still follow or closely resemble the one shown in Fig. 2(a), which directly utilizes or further aggregates these local features for feature matching. In contrast, our approach utilizes an additional decoder specifically for registration. Though simple, this resolves the conflict between the requirements of the two tasks. The incorporation of dense local features has the potential to enhance registration performance by establishing more correct matches, while maintaining the efficient generation of global descriptors. However, ensuring the match quality in an increased search space can be difficult and time-consuming. We thereby have implemented a sparse-to-dense matching approach for reliable and fast registration. By exploiting different types of features and a multihead network, we concurrently address the disparities between loop closing and relocalization tasks while leveraging their shared characteristics to unify them within a single framework. More detailed description of the proposed network following our framework is introduced in the next section.

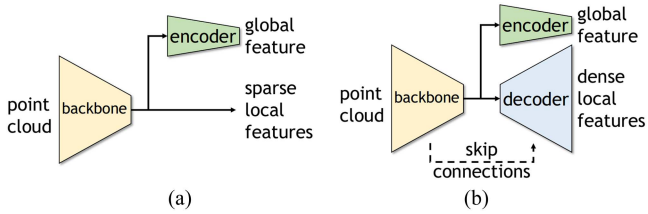

Fig. 2. Comparison of common frameworks and our framework for similarity evaluation and registration tasks. Our framework introduces a decoder to generate denser local features for registration task. (a) Common framework. (b) Our framework.

IV. Loop Closing and Relocalization Network

To realize our proposed framework, we design a novel multihead network, named LCR-Net. As shown in Fig. 3, it consists of a keypoint detection module (Section IV-A) to extract keypoints from the raw point cloud, a global description head (Section IV-B) for global descriptor generation, and a dense point matching head (Section IV-C) for local feature generation and matching. The devised loss function and the training strategy of our approach are detailed in Section IV-D and Section IV-E, respectively.

A. Keypoint Detection Module

The keypoint detection module aims to downsample the point cloud into sparse keypoints for further processing in two heads.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

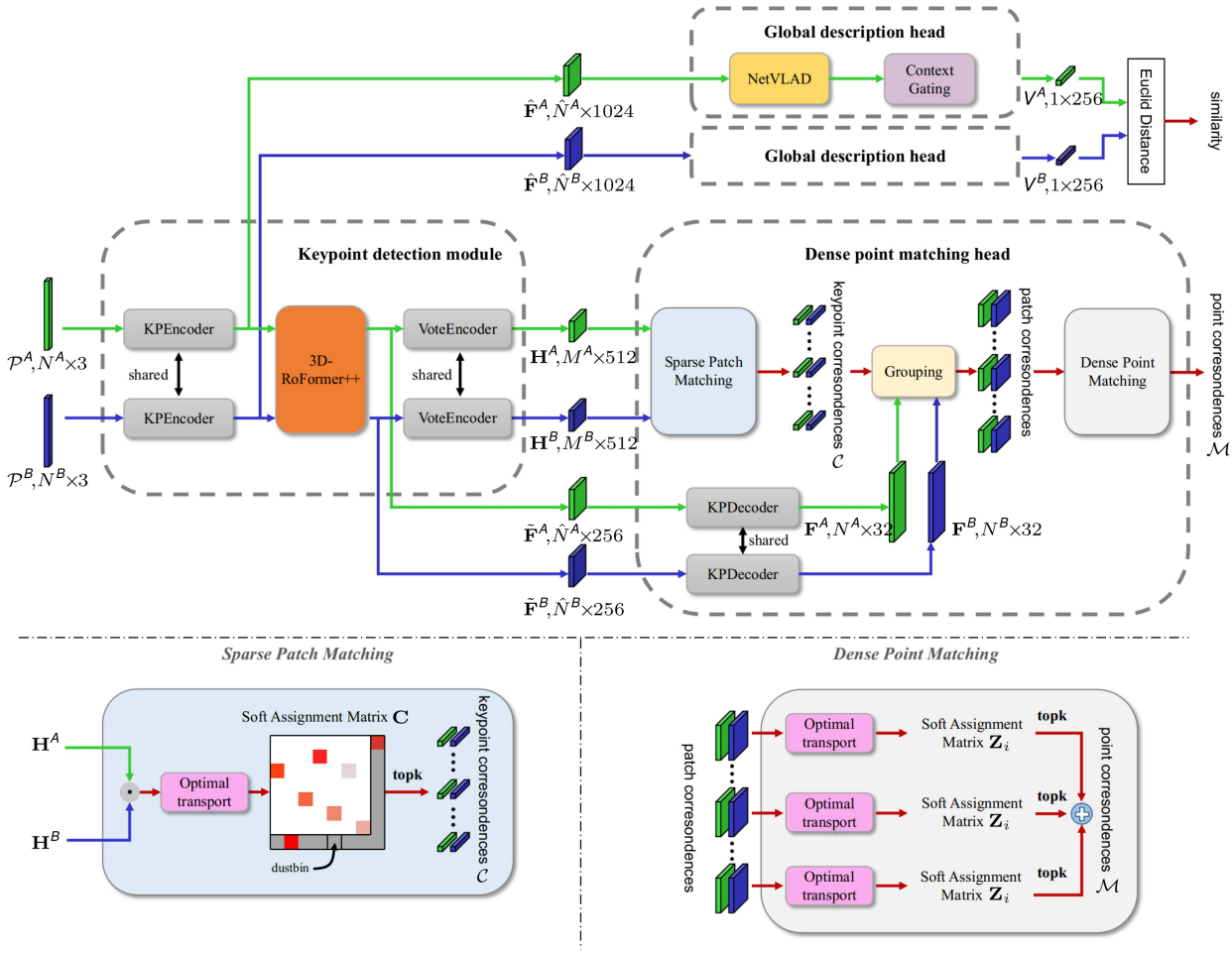

Fig. 3. Pipeline overview. LCR-Net consists of three main components: A keypoint detection module, a global description head, and a dense point matching head. The keypoint detection module extracts three types of features for further processing in two heads. The global description head generates a global descriptor for fast candidate retrieval. The dense point matching head exploits a sparse-to-fine approach to establish dense point matching for 6-DoF pose estimation.

In this work, we utilize KPEncoder [53] as the starting point for extracting the features. KPEncoder comprises a series of downsampling and kernel point-based convolution (KPConv) blocks, enabling hierarchical encoding of the point cloud into the uniformly distributed keypoints with descriptors $[\hat{\mathcal{P}}|\hat{\mathbf{F}}]$. These features provide sufficient information about the overall structure of the point cloud and are well-suited for input into the global description head. However, they suffer from a lack of information exchange between two scans. Moreover, the uniformly sampled keypoints can not satisfy the demand for accurate registration due to their limited repeatability and saliency. To address these limitations, we introduce the *3D-RoFormer++* to reason about contextual information in both point clouds, and the *VoteEncoder* that generates new keypoints nearby significant regions based on the enhanced features. Each component is detailed below.

3D-RoFormer++: Our previous work introduced the 3D-RoFormer [41] for lightweight relative pose-aware contextual aggregation. In this article, we have brought it to maturity and present the 3D-RoFormer++ by providing valuable translational invariance and enhanced feature representation performance. The 3D-RoFormer is built upon the vanilla Transformer [19]. For a point $\boldsymbol{p}_i^Q$ with its feature $\boldsymbol{h}_i^Q$ in the query point cloud $\boldsymbol{Q}$ and all the points in the source point cloud $\boldsymbol{S}$, the Transformer computes the query $\boldsymbol{q}_i$, key $\boldsymbol{k}_j$, and value $\boldsymbol{v}_j$ feature maps with linear projections. In addition to the contextual features, 3D-RoFormer encodes the position $\hat{\boldsymbol{p}}_i \in \mathbb{R}^3$ into the rotary embedding $\boldsymbol{\Theta}_i = [\theta_1, \theta_2, \ldots, \theta_{d/2}] \in \mathbb{R}^{\frac{d}{2}}$ defined as follows:

$$\boldsymbol{\Theta}_i = f_{\text{rot}}(\hat{\boldsymbol{p}}_i)) = 2\pi \cdot \text{sigmoid}(\text{MLP}(\hat{\boldsymbol{p}}_i)). \tag{1}$$

By treating each element in $\boldsymbol{\Theta}_i$ as a rotation in a 2-D plane, it can be converted to a rotation matrix $\boldsymbol{R}_{\boldsymbol{\Theta}_i} \in \mathbb{R}^{d\times d}$ given by

$$\boldsymbol{R}_{\boldsymbol{\Theta}_i} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & \cdots & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos\theta_{\frac{d}{2}} & -\sin\theta_{\frac{d}{2}} \\ 0 & 0 & \cdots & \sin\theta_{\frac{d}{2}} & \cos\theta_{\frac{d}{2}} \end{bmatrix}. \tag{2}$$

Applying $\boldsymbol{R}_{\boldsymbol{\Theta}_i}$ and $\boldsymbol{R}_{\boldsymbol{\Theta}_j}$ to query $\boldsymbol{q}_i$ and key $\boldsymbol{k}_j$, respectively, in self-attention operation, the rotary self-attention in

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

3D-RoFormer can be written as follows:

$$\alpha''_{ij} = \underset{j}{\text{softmax}}((R_{\Theta_i}\boldsymbol{q}_i)^{\mathsf{T}} R_{\Theta_j}\boldsymbol{k}_j) \tag{3}$$

$$\tilde{\boldsymbol{f}}_i = \sum_{j=1}^{|\hat{\mathcal{P}}|} \alpha''_{ij}\boldsymbol{v}_j. \tag{4}$$

Equation (3) can be further written as follows:

$$\alpha''_{ij} = \underset{j}{\text{softmax}}\left(\boldsymbol{q}_i^{\mathsf{T}} R_{\Theta_i}^{\mathsf{T}} R_{\Theta_j}\boldsymbol{k}_j\right) = \underset{j}{\text{softmax}}(\boldsymbol{q}_i^{\mathsf{T}} R_{\Theta_j - \Theta_i}\boldsymbol{k}_j). \tag{5}$$

The important advantage of 3D-RoFormer is that it explicitly encodes the relative geometric information neatly without requiring extra-large storage memory for relative position embedding. As in (5), relative "rotation" $\boldsymbol{\Theta}_j - \boldsymbol{\Theta}_i$ is naturally incorporated into the calculation and then fused with the output feature $\tilde{\boldsymbol{f}}_i$ in (4). Furthermore, if the mapping function f_{rot} is linear, we can further derive the difference in rotational embeddings by the following:

$$\boldsymbol{\Theta}_j - \boldsymbol{\Theta}_i = f_{\text{rot}}(\hat{\boldsymbol{p}}_j - \hat{\boldsymbol{p}}_i). \tag{6}$$

This leads to a very important property for keypoint detection, which is translation-invariance. However, designing a f_{rot} that provides good rotary feature representation while maintaining linearity is challenging. To ensure the ability of rotary representation, the rotary embedding in the original 3D-RoFormer, as shown in (1), sacrifices linearity for rotary representation, leading to reduced generalization performance.

Based on this insight, we improve our 3D-RoFormer by adopting a learning-based linear mapping function defined as follows:

$$\boldsymbol{\Theta}_i = \text{Linear}(\hat{\boldsymbol{p}}_i) \tag{7}$$

with a boundary penalty loss (Section IV-D) as an auxiliary loss to supervise the network actively learning effective rotary representations. With this, 3D-RoFormer++ largely enhances the output features $\tilde{\mathbf{F}}^A$ and $\tilde{\mathbf{F}}^B$ for point matching by interleaving the rotary self-attention and cross-attention for l times.

The enhanced features $\tilde{\mathbf{F}}$ possess geometric and contextual information between two point clouds, which is then extended to dense features for further processing in dense point matching head, as detailed in Section IV-C. Nevertheless, the uniform sampling nature makes these features less salient and discriminative. We, therefore, propose the VoteEncoder.

VoteEncoder: To steer the evenly sampled features $[\hat{\mathcal{P}}|\tilde{\mathbf{F}}]$ toward nearby salient areas and obtain more meaningful features conducive to registration tasks, we introduce the VoteEncoder for additional feature shifting and encoding.

We use a voting module [41], [54] to estimate the geometric offset from the uniformly sampled keypoints to the proposal keypoints $\mathcal{S}$, i.e., $\Delta\mathbf{P} = \text{Vote}(\tilde{\mathbf{F}})$ and $\mathcal{S} = \hat{\mathcal{P}} + \Delta\mathbf{P}$. The voting module comprises a collection of multilayer perceptrons (MLPs). Despite its simplicity, this module produces meaningful offsets (see Fig. 4), utilizing the features from our 3D-RoFormer++. After voting, multiple proposals emerge clustered within salient areas of the point cloud. In our previous work [41], we randomly select a single proposal from each cluster as the final keypoints, which might be unstable and result in substantial information loss. In this work, we adopt a more rational and effective approach. We estimate the centroids of each cluster as the keypoints and subsequently aggregate information from each cluster to generate new features that serve as descriptors for the keypoints. The process of predicting centroids is straightforward yet efficient, without the need for additional sampling strategies. It clusters all the proposals into various patches and predicts the centers, detailed in Algorithm 1. To aggregate the descriptors $\mathbf{H}$ for each center point $\hat{\mathcal{S}}$, we employ a KPConv module that performs kernel-based convolution after finding nearest neighbors of $\hat{\mathcal{S}}$ in $[\hat{\mathcal{P}}|\tilde{\mathbf{F}}]$. We use a larger search range than that used in Algorithm 1 to incorporate more related context near the keypoints.

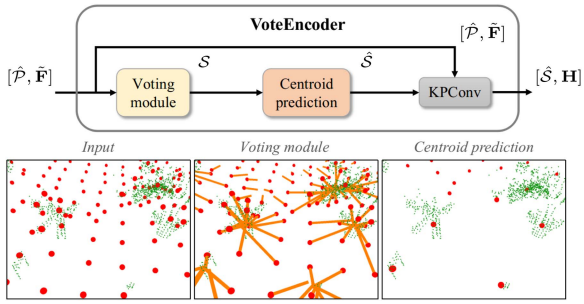

Fig. 4. VoteEncoder takes sparse features as input and generates offset from each keypoint to its nearest significant region. The final keypoints are estimated using centroid prediction algorithm, and their features are aggregated by KPConv. The ground points are removed from the visualization for clarity.

Algorithm 1: Centroid Prediction.

Input: proposal set $\mathcal{S}$, nearest neighbor search range d
Output: center point set $\hat{\mathcal{S}}$
1: **for** all $\boldsymbol{s}_i \in \mathcal{S}$ **do**
2: **if** ISLABELED($\boldsymbol{s}_i$) is False **then**
3: $\mathcal{N}_i \leftarrow$ NEARESTNEIGHBOR($\boldsymbol{s}_i, \mathcal{S}, d$)
4: $\hat{\boldsymbol{s}}_i \leftarrow$ MEAN($\mathcal{N}_i$) ▷ Get centroid point
5: $\hat{\mathcal{S}}$.APPEND($\hat{\boldsymbol{s}}_i$)
6: **for** all $\boldsymbol{s}_j \in \mathcal{N}_i$ **do**
7: ISLABELED($\boldsymbol{s}_j$) $\leftarrow$ True
8: **return** $\hat{\mathcal{S}}$

Unlike the object detection [54], [55], where the object center can serve as a well-defined reference for supervising point shifts, our case lacks a readily available ground truth center for the significant areas, primarily due to the challenge in precisely defining the significance. Therefore, we train the matched keypoints to move closer to each other instead, which indirectly accomplishes our objective. In practice, the offset $\Delta\mathbf{P}$ is limited to a certain range to maintain an even distribution of keypoints throughout the point cloud. This prevents the keypoints from being only concentrated in significant areas while also avoiding potential degeneracy.

In sum, our keypoint detection module offers various options for the sparse features, including uniformly sampled features $\hat{\mathbf{F}}$ from KPEncoder [53], enhanced features $\tilde{\mathbf{F}}$ from

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

3D-RoFormer++ and voted features $\mathbf{H}$ from VoteEncoder. Uniformly sampled features $\hat{\mathbf{F}}$ are distributed evenly throughout the entire point cloud, enabling fast extraction and the capacity to represent the entire point cloud comprehensively. These features are utilized for the global description head, as detailed in Section IV-B. On the other hand, enhanced features $\tilde{\mathbf{F}}$, built upon the uniformly sampled features $\hat{\mathbf{F}}$, incorporate geometric information and the correlation between two scans. These features will be decoded into dense features, propagating the aforementioned advantages to them. Finally, voted features $\mathbf{H}$ exhibit sensitivity and expressive capability for local salient regions while maintaining good coverage. These features will be employed for initial sparse matching, which will then be extended to dense matching for accurate registration. Both enhanced features and voted features will be employed for the dense point matching head detailed in Section IV-C.

B. Global Description Head

The objective of the global description head is to condense sparse features into a single global feature for fast candidate retrieval. We adopt the features derived from the KPEncoder as the input to our global description head. We choose a widely applied simple method NetVLAD [14] to compress the features. It uses k-means clustering and aggregates all the features into K cluster features $\mathbf{FR} \in \mathbb{R}^{K\times\tilde{d}}$. Then, a simple MLP compresses $\mathbf{FR}$ into a single descriptor $\boldsymbol{X} \in \mathbb{R}^{G}$.

Based on NetVLAD, we use the context gating module [5] to re-evaluate the weights of each channel of feature $\boldsymbol{X}$ based on the self-attention mechanism and further enhances it to obtain the final global descriptor $\boldsymbol{V} \in \mathbb{R}^{G}$, given as follows:

$$\boldsymbol{V} = \mathrm{CG}(\boldsymbol{X}) = \sigma(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}) \otimes \boldsymbol{X} \tag{8}$$

where σ is the sigmoid activation function, $\otimes$ is elementwise multiplication, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable weights and offsets.

The design of the global description head is straightforward yet remarkably effective, surpassing all baseline methods in our experiments. Its simplicity is also particularly important because LiDAR SLAM requires quick and accurate retrieval for real-time canditate retrieval, which narrows down the computational scope for subsequent fine 6-DoF pose estimation.

C. Dense Point Matching Head

Once identifying the candidates, we leverage the dense point matching head to establish correspondences and subsequently recover precise 6-DoF pose estimation. The features obtained from our keypoint detection module $[\hat{\mathcal{S}}|\mathbf{H}]$ are sufficient for ensuring dependable point cloud registration. However, there are two factors that impact the accuracy of the final 6-DoF pose estimation. First, despite VoteEncoder improving keypoint locations, there might still be noticeable distances between matched sparse features. These gaps can lead to errors that restrict the overall accuracy. Second, due to the sparse characteristics of these features, there might not be adequate feature matches to fully rectify errors arising from mismatches. Considering these limitations, we employ a two-step matching approach [39], [40]. Initially, we identify sparse yet dependable keypoint matches, and then we extend these point-to-point matches to patch-to-patch matches. By utilizing neighbor consistency, we enhance these patch matches into dense point matches, ensuring more precise and reliable registration.

Sparse keypoint matching: We conduct sparse matching between $[\hat{\mathcal{S}}^{A}|\mathbf{H}^{A}]$ and $[\hat{\mathcal{S}}^{B}|\mathbf{H}^{B}]$. We compute a matching score matrix $\mathbf{C} \in \mathbb{R}^{|\hat{\mathcal{S}}^{A}|\times|\hat{\mathcal{S}}^{B}|}$ between $\mathbf{H}^{A}$ and $\mathbf{H}^{B}$ as $\mathbf{C} = \mathbf{H}^{A}(\mathbf{H}^{B})^{\mathsf{T}}/\sqrt{d_c}$, where d_c refers to the feature dimension of $\mathbf{H}$. To handle nonmatched points, we append a "dustbin" [56] row and column for $\mathbf{C}$ filled with a learnable parameter $\alpha \in \mathbb{R}$. The Sinkhorn algorithm [57] is then used to solve the soft assignment matrix. It iteratively performs normalization along rows and columns. At the t iteration, the score matrix is updated as follows:

$$^{(t)}\mathbf{C}'_{ij} = {}^{(t)}\mathbf{C}_{ij} - \log \sum_{j} e^{{}^{(t)}\mathbf{C}_{ij}} \tag{9}$$

$$^{(t+1)}\mathbf{C}_{ij} = {}^{(t)}\mathbf{C}'_{ij} - \log \sum_{i} e^{{}^{(t)}\mathbf{C}'_{ij}}. \tag{10}$$

After T iterations, we use the solution as the soft assignment matrix $\hat{\mathbf{C}} = {}^{(T)}\mathbf{C}$. We choose the largest N_c entries as the keypoint correspondences as follows:

$$\mathcal{C} = \left\{ \left(\hat{\boldsymbol{s}}^{A}_{x_i}, \hat{\boldsymbol{s}}^{B}_{y_i} \right) \mid (x_i, y_i) \in \text{Top-k}_{x,y}\left(\hat{\mathbf{C}}\right) \right\}. \tag{11}$$

Patch grouping: To achieve dense point matches from sparse keypoint matches, we expand correspondences between keypoints to encompass overlaps between their respective neighborhood patches and subsequently leverage these patches to identify more point matches.

For each keypoint $\hat{\boldsymbol{s}}_i$, we construct a local patch $\mathcal{G}_i$ using a point-to-node strategy [36], where each point is assigned to its nearest keypoint. Based on the grouped point patch, we can now extend each keypoint match $(\hat{\boldsymbol{s}}^{A}_{x_i}, \hat{\boldsymbol{s}}^{B}_{y_i})$ to its corresponding patch match $(\mathcal{G}^{A}_{x_i}, \mathcal{G}^{B}_{y_i})$.

Dense point matching: We then generate more point matches from the sparse patch matches. We leverage the KPDecoder [53] to recover point-level descriptors $\mathbf{F}$ from enhanced keypoint features $[\hat{\mathcal{P}}|\tilde{\mathbf{F}}]$. For each keypoint correspondence $(\hat{\boldsymbol{s}}^{A}_{x_i}, \hat{\boldsymbol{s}}^{B}_{y_i})$, we compute a match score matrix $\mathbf{O}_i \in \mathbb{R}^{|\mathcal{G}^{A}_{x_i}|\times|\mathcal{G}^{B}_{y_i}|}$ of their corresponding patches $\mathcal{G}^{A}_{x_i}$ and $\mathcal{G}^{B}_{y_i}$, given by $\mathbf{O}_i = \mathbf{F}^{A}_{x_i}(\mathbf{F}^{B}_{y_i})^{\mathsf{T}}/\sqrt{d_f}$, where d_f refers to the feature dimension of $\mathbf{F}$. Same with our sparse keypoint matching module, we append a learnable "dustbin" row and column for $\mathbf{O}_i$ to handle nonmatched points and use the Sinkhorn algorithm to solve the soft assignment matrix $\mathbf{Z}_i \in \mathbb{R}^{(|\mathcal{G}^{A}_{x_i}|+1)\times(|\mathcal{G}^{B}_{y_i}|+1)}$. Unlike works [39], [40] that drop the dustbin and recover the assignment by comparing the soft assignment score with a hand-tuned threshold, we directly find the largest entry both row and columnwise on $\mathbf{Z}_i$ to recover the assignment $\mathcal{M}_i$ given as follows:

$$\mathcal{M}_i = \Big\{ \big(\mathcal{G}^{A}_{x_i}(m), \mathcal{G}^{B}_{y_i}(n)\big) \big| (m,n) \in \text{toprow}_{m,n}\left(\mathbf{Z}^{i}_{1:M_i,1:(N_i+1)}\right) \Big\}$$

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

$$\cup \left\{ (\mathcal{G}^{A}_{x_i}(m), \mathcal{G}^{B}_{y_i}(n)) | (m,n) \in \text{topcolumn}_{m,n}\left(\mathbf{Z}^{i}_{1:(M_i+1),1:N_i}\right)\right\} \quad (12)$$

where $\text{toprow}_{m,n}()$ and $\text{topcolumn}_{m,n}()$ return the index of the top value rowwise and columnwise, respectively. A point is either assigned to points in the matched patch or to the dustbin. By doing this, we do not need manual tuning but require a discriminative assignment matrix, which can be obtained by using our proposed loss function as detailed in Section IV-D. Note that a point is not strictly assigned to a single point in our approach, as the strict one-to-one point correspondences do not hold in practice due to the sparsity nature of the LiDAR scans. Instead, we trust and keep the assignment results from both sides, i.e., matches from query to source and vice versa. This results in extensively more point matches while maintaining a high IR, which benefits the transformation estimation. The final correspondences are the combination of points matches from all patches as follows:

$$\mathcal{M} = \bigcup_{i=1}^{N_c} \mathcal{M}_i. \quad (13)$$

Local-to-global registration: We employ local-to-global registration (LGR) proposed in [40] for fast pose estimation. It is a hypothesize-and-verify approach specifically proposed for matching methods following a sparse-to-dense manner. For each matched patch, LGR solves a transformation $\{\boldsymbol{R}_i, \boldsymbol{t}_i\}$ based on its dense point matches using weighted singular value decomposition (SVD) [8], where the soft assignment value in $\mathbf{Z}^i$ serves as the weight. After obtaining the transformations for all matched patches, LGR selects the transformation that has the most inliers among all dense point matches, calculated as follows:

$$\boldsymbol{R}, \boldsymbol{t} = \max_{\boldsymbol{R}_i, \boldsymbol{t}_i} \sum_{(\boldsymbol{p}^{A}_{x_j}, \boldsymbol{p}^{B}_{y_j}) \in \mathcal{M}} \left[\!\left[\| \boldsymbol{R}_i \cdot \boldsymbol{p}^{A}_{x_j} + \boldsymbol{t}_i - \boldsymbol{p}^{B}_{y_j} \|_2^2 < \tau_a \right]\!\right] \quad (14)$$

where $[\![\bullet]\!]$ is an indicator function for which the statement is true. Finally, it solves the final transformation $\boldsymbol{R}, \boldsymbol{t}$ by solving another weighted SVD on surviving inliers for N_r times.

LGR significantly reduces the number of iterations compared with RANSAC [29], achieving a substantial speed advantage with about 30 times faster in our experiments. However, the performance of LGR, particularly its robustness, can be heavily influenced by the quality of sparse patch matching. We significantly improve the matching quality of sparse patches through the powerful feature aggregation module 3D-RoFormer++ and the feature detection module VoteEncoder, achieving performance comparable with or surpassing RANSAC's accuracy and robustness.

D. Loss Function

To effectively guide our network in accomplishing various tasks, we construct our loss function with five components: the keypoint detection loss L_{s}, the boundary penalty loss for keypoint detection module, the triplet loss L_{t} for global description head, the sparse match loss L_{c}, and the dense match loss L_{f} for dense point matching head.

Keypoint detection loss: The keypoint detection loss consists of two parts $L_s = L_{s1} + L_{s2}$. The first part L_{s1} is designed to guide the corresponding keypoints from two point clouds close to each other lying within the significant region

$$L_{\text{s1}} = \sum_{i=1}^{|\mathcal{S}^A|} \min_{\boldsymbol{s}^B_j \in \mathcal{S}^B} \|\boldsymbol{s}^A_i - \boldsymbol{s}^B_j\|_2^2 + \sum_{i=1}^{|\mathcal{S}^B|} \min_{\boldsymbol{s}^A_j \in \mathcal{S}^A} \|\boldsymbol{s}^B_i - \boldsymbol{s}^A_j\|_2^2. \quad (15)$$

Supervised by L_{s1}, we find that the keypoints tend to move to their nearest significant regions to indirectly minimize the distance between keypoint pairs.

The second part L_{s2} is designed to make the keypoints close to the real measurement points. It minimizes the distance between the keypoint with its closest point as follows:

$$L_{s2} = \sum_{i=1}^{|\mathcal{S}^A|} \min_{\boldsymbol{p}^A_j \in \mathcal{P}^A} \|\boldsymbol{s}^A_i - \boldsymbol{p}^A_j\|_2^2 + \sum_{i=1}^{|\mathcal{S}^B|} \min_{\boldsymbol{p}^B_j \in \mathcal{P}^B} \|\boldsymbol{s}^B_i - \boldsymbol{p}^B_j\|_2^2. \quad (16)$$

Boundary penalty loss: To guide the 3D-RoFormer++ learn a general rotary embedding representation, we add a boundary penalty loss to force the value of the rotary embedding $\boldsymbol{\Theta}$ to lie between $[-\pi, \pi]$ as follows:

$$L^i_{\text{p}} = \frac{1}{M_i} \sum_{m=1}^{M_i} [\text{abs}(\boldsymbol{\Theta}) - \pi]_+, \; [\bullet]_+ = \max(\bullet, 0). \quad (17)$$

Triplet loss: We use the triplet loss [58] to train the global description head. Following [25], [26], and [27], we use overlap to define positive and negative samples for each scan, where positive samples have overlaps larger than 0.3 and otherwise defined as negatives. For each triplet, we use one query scan, $N_p = 6$ positive scans and $N_n = 6$ negative scans. The triplet loss is calculated as follows:

$$L_{\text{t}}(\boldsymbol{V}_q, \{\boldsymbol{V}_p\}, \{\boldsymbol{V}_n\}) = N_p \left[\alpha + \max_p (d(\boldsymbol{V}_q, \boldsymbol{V}_p)) - \frac{1}{N_n} \sum_{N_n} (d(\boldsymbol{V}_q, \boldsymbol{V}_n)) \right]_+. \quad (18)$$

Sparse match loss: We utilize a gap loss [38] to learn a discriminative soft assignment matrix $\mathbf{C}$ for sparse keypoint matching. The ground truth of assignment matrix $\mathbf{P} \in \{0,1\}^{(M+1)\times(N+1)}$ is generated based on the overlap ratio between the patches, where $M = |\hat{\mathcal{S}}^A|$ and $N = |\hat{\mathcal{S}}^B|$ are the number of keypoints. Two patches are matched when they share at least 10% overlap. A patch is assigned to the dustbin when it has no match pair. We also generate a negative assignment matrix $\bar{\mathbf{P}} \in \{-\inf, 1\}^{(M+1)\times(N+1)}$, where 1 represents two patches are not overlapped and $\inf$ represents the value will not be involved in the calculation of loss. Then, the gap loss is calculated as follows:

$$L_{\text{c}} = f_{\text{gap}}(\mathbf{C}, \mathbf{P}, \bar{\mathbf{P}}) = \frac{1}{M} \sum_{m=1}^{M} \log \left(\sum_{n=1}^{N+1} (-r_m + \mathbf{C}_{m,n} \bar{\mathbf{P}}_{m,n} + \eta)_+ + 1 \right)$$

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

$$+\frac{1}{N}\sum_{n=1}^{N}\log\left(\sum_{m=1}^{M+1}(-c_n+\mathbf{C}_{m,n}\bar{\mathbf{P}}_{m,n}+\eta)_+ +1\right) \tag{19}$$

where $r_m=\max_m(\mathbf{C}_{m,n}\mathbf{P}_{m,n})$ and $c_n=\max_n(\mathbf{C}_{m,n}\mathbf{P}_{m,n})$ refer to the soft assignment value for the hardest true match of mth keypoint in $\hat{\mathcal{S}}^A$ and nth keypoint in $\hat{\mathcal{S}}^B$, respectively, and $(\bullet)_+=\max(\bullet,0)$.

Dense match loss: The dense match loss is calculated over all the matched patches. For each matched patch pair $\{\mathcal{G}^A_{x_i},\mathcal{G}^B_{y_i}\}$, we generate its ground truth positive correspondences matrix $\mathbf{M}^i\in\{0,1\}^{(M_i+1)\times(N_i+1)}$ and negative matrix $\bar{\mathbf{M}}^i\in\{10^{12},1\}^{(M_i+1)\times(N_i+1)}$ with a distance threshold τ, where $M_i=|\mathcal{G}^A_{x_i}|$ and $N_i=|\mathcal{G}^B_{x_i}|$. A point pair is positive when the distance is below τ and is negative when it exceeds 2τ. To learn a discriminative soft assignment matrix, we also calculate a gap loss $L^i_{\mathrm{f}}=f_{\mathrm{gap}}(\mathbf{Z}^i,\mathbf{M}^i,\bar{\mathbf{M}}^i)$ for patch correspondence's soft assignment matrix $\mathbf{Z}^i$. The final fine match loss is the average over all the matched patch pairs $L_{\mathrm{f}}=\frac{1}{2|\mathcal{M}|}\sum_{i=1}^{|\mathcal{M}|}L^i_{\mathrm{f}}$.

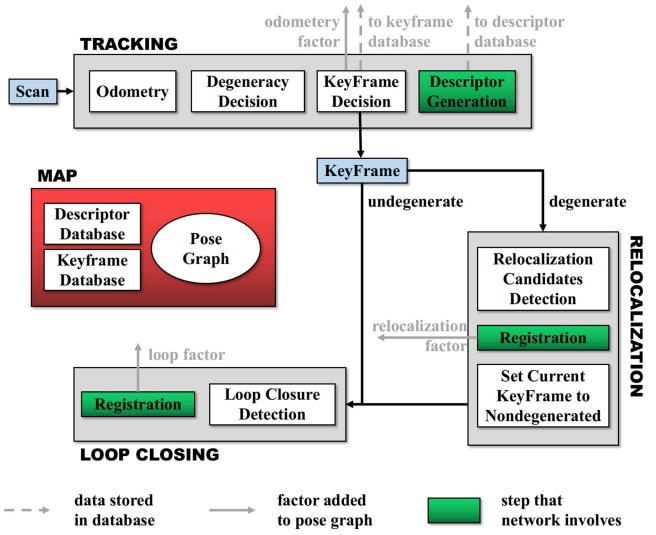

Fig. 5. Overview of the laser SLAM system integrated with LCR-Net, showing key steps executed by the tracking, relocalization and loop closing threads. LCR-Net serves for relocalization and loop closing.

E. Training Strategy

We seek a training strategy to stimulate the potential of each head with limited computing resources. As a result, a two-stage training strategy is employed. We first train the keypoint detection module and the dense point matching head for registration. Then, we exclusively train the global description head for candidate retrieval for the following reasons. First, the input for the training of two heads differs. The training of the global description head requires a minimum of three scans: an anchor, a positive, and a negative. Conversely, training the dense point matching head only requires two overlapped scans. Including additional input does not benefit the training of the dense point-matching head but consumes a significant amount of memory. Second, for candidate retrieval, a higher batch size typically results in better performance. By freezing the keypoint detection module and dense point matching head, we can use the preextracted features for input, thereby preserving substantial memory for expanding the batch size.

However, using preextracted features prevents the implementation of data augmentation techniques, thereby limiting performance. To address this problem, we utilize a training strategy, which we refer to as semionline. In this approach, we utilize offline preextracted features for both positive and negative samples while generating features for the anchor online. This allows for applying data augmentation on the anchor. Since the anchor participates in loss calculations with all positive and negative samples, this can be the most efficient way to implement data augmentation.

In sum, our two-stage training strategy first trains the network in the registration task, and then exclusively trains the global description head semionline for the candidate retrieval task. This strategy offers several advantages. First, it allows us to utilize larger batch sizes and sample quantities during training for the candidate retrieval task. Second, the semionline approach enables applying data augmentation techniques. Lastly, this strategy provides great convenience in training as we only need to select the best model in the registration task for the second-stage training, followed by selecting the best model in the candidate retrieval task. These advantages contribute to the noteworthy enhanced performance.

We train our LCR-Net on 4 NVIDIA RTX 3090 GPUs. The network is trained using the Adam optimizer [59] with an initial learning rate as 10^{-4}, which undergoes exponential decay by 0.05 every 4 epochs. When training the dense point matching head, we use a batch size of one. When training the global description head, we set the batch size to six. We also apply the same data augmentation techniques as in [5].

V. LCR-Net Based LiDAR SLAM System

We integrate our LCR-Net into a SLAM system for relocalization and loop closing. We use an incremental smoothing and mapping (iSAM2) [60] based pose-graph optimization (PGO) framework to manage and optimize the global pose graph. The system comprises three threads that run in parallel: tracking, relocalization, and loop closing, as shown in Fig. 5.

A. Tracking

The tracking thread is responsible for tracking the LiDAR pose of every scan, i.e., odometry. It also decides whether the current odometry estimation is degenerated and when to insert a new keyframe. The process of odometry can be formulated as a state estimation problem, given as follows:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}}\, f^2(\boldsymbol{x}) \tag{20}$$

where $\boldsymbol{x}=[\boldsymbol{R},\boldsymbol{t}]$ is the state vector. Given an initial guess of $\boldsymbol{x}$, most nonlinear optimization methods solve the function by computing the Jacobian matrix of f w.r.t $\boldsymbol{x}$ as follows:

$$\boldsymbol{J}=\delta f(\boldsymbol{x})/\delta\boldsymbol{x} \tag{21}$$

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

and iteratively adjust x utilizing J until convergence. The tracking thread can be implemented through LiDAR odometry, such as LiDAR odometry and mapping (LOAM) [61]. For degeneracy evaluation, we adopt the criterion proposed in [46], which considers the problem degenerates if the smallest eigenvalue of matrix $J^T J$ falls below a certain threshold. In practice, the threshold is conservatively set to ensure that all degeneracy can be detected. Though this leads to increased computational load, it is necessary since even a single instance of severe degeneracy could inflict significant damage on the system.

LiDAR scans are keyframes if the robot has moved beyond a predefined threshold from the last saved keyframe. However, when odometry performance deteriorates, relying solely on the previous travel distance to establish keyframes becomes unreliable. To address this, we introduce additional scans as keyframes based on specific conditions as follows.

1) The first degenerated scan follows nondegenerated scans.
2) The first nondegenerated scan follows a degenerated scan, provided the degenerated scan is not chosen as a keyframe.
3) Every t-second scan within a continuous sequence of degenerated scans.

The keyframes selected according to these conditions are categorized as "degenerated," as poses estimated between these keyframes may include degenerated ones. Finally, for each keyframe, we use LCR-Net to generate a global descriptor. The similarity between our generated global descriptors is evaluated based on the Euclidean distance in the feature space. For fast retrieving, we utilize the FAISS library [62] for descriptor database management and search.

B. Relocalization

The relocalization processes every "degenerated" keyframe, generating more accurate pose estimation to replace the unreliable odometry output. We select the most recent "nondegenerated" keyframe for each degenerated keyframe as the candidate and match it with the current keyframe using LCR-Net. For real-time efficiency, we utilize the rapid estimator LGR for pose estimation. Our experimental results show that when working with LCR-Net, LGR offers a significantly faster processing speed of nearly 30 times than RANSAC, while producing comparable accuracy and robustness in registration. To assess whether the relocalization is successful, we calculate a current pose estimation reliability score by averaging the assignment score of all the found correspondences. The relocalization succeeds if the reliability score surpasses a threshold ρ_r. Once the calculated pose is included in the graph optimization, the label assigned to the current keyframe is updated to "nondegenerated." In most cases, this allows for recovery of the pose tracking. Otherwise, we retrieve the most similar candidate with descriptor distance below a threshold ρ_s from the database and attempt relocalization.

C. Loop Closing

The loop closing thread functions by searching the loop candidate and subsequently estimating pose as a new node added to the pose graph. Specifically, we retrieve the candidate from the database that has the most similar descriptor for each new keyframe while excluding the 100 most recent keyframes. If the distance is below a threshold ρ_s, we set the retrieved keyframe as the loop and estimate the relative pose using LCR-Net with LGR. Finally, the iSAM2-based PGO is performed to achieve global consistency.

VI. Experimental Evaluation

We conduct experiments to demonstrate the efficacy of our proposed LCR-Net in addressing loop closing and relocalization for online LiDAR SLAM. We derive three setups from these challenges: candidate retrieval, closed-loop point cloud registration, and continuous relocalization. In candidate retrieval experiments (Section VI-B), we examine the capability of LCR-Net in accurately retrieving the appropriate candidates from the previous map. In closed-loop point cloud registration (Section VI-C) and continuous relocalization experiments (Section VI-D), we assess the ability of LCR-Net in successfully registering point clouds during loop situations as well as continuous scenarios with low overlap. We also evaluate the runtime of our approach for candidate retrieving and point cloud registration (Section VI-E). We then evaluate our LCR-Net enhanced SLAM in multiple real-world scenes (Section VI-F). Finally, we conduct ablation studies on the network design (Section VI-G) and provide valuable insights (Section VI-H).

A. Experimental Setup

We evaluate LCR-Net and compare it with the SOTA methods on multiple publicly available datasets, including KITTI odometry [63], KITTI-360 [64], Apollo-SouthBay [65], Ford Campus [66], and Mulran [67] datasets. These datasets provide LiDAR scans collected in various environments in multiple countries with the corresponding ground-truth poses.

We use the two-stage training strategy described in Section IV-E: Pretrain the network with dense point matching head in the registration task, and then linear probe the global description head for training in the candidate retrieval task. For a fair comparison, we have trained two models denoted as *LCR-Net*† and *LCR-Net*$^\diamond$, using different dataset splittings that follow the existing baselines for specific tasks: *LCR-Net*† is pretrained in closed-loop point cloud registration task, whereas *LCR-Net*$^\diamond$ is pretrained in continuous relocalization task. Both models are then trained in the candidate retrieval task.

Aiming to address all three tasks in a single model, we have also trained another model denoted as *LCR-Net*. In particular, we divide the KITTI odometry dataset into the following: sequences 01 and 03–07 for training, sequence 02 for validation, and sequences 00, 08–10 for testing. The other datasets are all used to test the models' generalization capabilities. For the registration task pretrain, we select point cloud pairs with a mix of continuous point cloud pairs with the distance varying from 0–10 m and closed-loop point cloud pairs with an overlap ratio exceeding 0.3 [25]. In the candidate retrieval task training, we use all point cloud pairs with an overlap ratio exceeding 0.3. We evaluate

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

LCR-Net on all subsequent tasks. As the model is trained on a different training set, we gray out the result of *LCR-Net*. This model is also used for the integration with our SLAM system.

B. Candidate Retrieval Performance

To validate the candidate retrieval performance, we follow Chen et al. [25], [26] and test our approach on the KITTI odometry and Ford Campus dataset. For a fair comparison, we follow the setup of Chen et al. [25], [26] and train *LCR-Net*† on KITTI odometry sequences 03–10 and validate it on KITTI odometry sequence 02. We also follow [25], [26], and [27] to calculate the overlap between scans. If the overlap value between two scans is larger than 0.3, they are selected as positive candidates; otherwise, they are considered negative. We also evaluate the *LCR-Net* trained on our dataset splittings as described in Section VI-A.

Metrics: We use four metrics to evaluate the performance of candidate retrieval as follows.

1) The area enclosed by the receiver operating characteristic curve and the coordinate axes.
2) Maximum F1 score (F1max), which is the highest F1 score at different threshold values.
3) Recall@1, which measures the recall when only the most similar candidate frame is selected.
4) Recall@1%, which measures the recall when the top 1% of the most similar candidate frames are selected.

Results: The baseline methods used in this experiment are SOTA place recognition methods. For Histogram [68], Scan Context [9], LiDAR-Iris [24], OverlapNet [25], PointNetVLAD [12], MinkLoc3D [15], and OverlapTranformer [26], we use the results reported in [26]. For LCDNet [5], MinkLoc3Dv2 [16], and LoGG3D-Net [17] we utilize its official implementation. As shown in Table I, LCR-Net and LCDNet achieve cutting-edge performance compared with current advanced methods. However, our method further boosts the baseline by a significant margin for the F1max metric while maintaining a leading position in Recall@1, Recall@1%, and area under the curve (AUC). It is worth noting that our global description head does not employ complex designs. The leading performance can be attributed to two key factors. First, the exceptional feature representation capabilities of our backbone play a fundamental role. Although the major part of the network is not directly involved in the calculation of the global descriptor, it still significantly influences the learning process of the backbone. A well-designed network facilitates better local feature representation, which, in turn, benefits the global description task, as can be seen from our ablation experiments. Second, our training strategy greatly contributes to the final performance. We employ a two-step training strategy that allows for a larger batch size, while still enabling the use of data augmentation techniques during the training of the global description head.

TABLE I
CANDIDATE RETRIEVAL RESULTS ON KITTI AND FORD CAMPUS

Dataset	Method	AUC	F1max	Recall @1	Recall @1%
KITTI	Histogram [68]	0.826	0.825	0.738	0.871
	Scan Context [9]	0.836	0.835	0.820	0.869
	LiDAR-Iris [24]	0.843	0.848	0.835	0.877
	PointNetVLAD [12]	0.856	0.846	0.776	0.845
	OverlapNet [25]	0.867	0.865	0.816	0.908
	MinkLoc3D [15]	0.894	0.869	0.876	0.920
	MinkLoc3Dv2 [16]	0.905	0.869	0.828	0.910
	LoGG3D-Net [17]	0.896	0.854	0.891	0.974
	OverlapTransformer [26]	0.907	0.877	0.906	0.964
	LCDNet [5]	0.933	0.883	0.915	0.974
	LCR-Net†	**0.945**	**0.907**	**0.926**	**0.980**
	LCR-Net	0.958	0.922	0.937	0.993
Ford Campus	Histogram [68]	0.841	0.800	0.812	0.897
	Scan Context [9]	0.903	0.842	0.878	0.958
	LiDAR-Iris [24]	0.907	0.842	0.849	0.937
	PointNetVLAD [12]	0.872	0.830	0.862	0.938
	OverlapNet [25]	0.854	0.843	0.857	0.932
	MinkLoc3D [15]	0.871	0.851	0.878	0.942
	MinkLoc3Dv2 [16]	0.931	0.854	0.923	0.976
	LoGG3D-Net [17]	0.924	0.867	0.906	0.962
	OverlapTransformer [26]	0.923	0.856	0.914	0.954
	LCDNet [5]	0.961	0.908	0.949	0.984
	LCR-Net†	**0.974**	**0.929**	**0.951**	**0.985**
	LCR-Net	0.972	0.920	0.932	0.987

LCR-Net† is trained on the same sets with baselines and LCR-Net is trained on our training sets.

C. Closed-Loop Point Cloud Registration Performance

We validate the registration performance of our method for closing the loop. We follow Cattaneo et al. [5] and test our method on KITTI odometry sequences 00 and 08 and KITTI-360 sequences 02 and 09. For a fair comparison, we follow [5] to train the *LCR-Net*† on the KITTI sequences 05–07 and 09, validating it on KITTI sequence 02. The point cloud pairs with ground truth pose distances less than 4 m and time intervals greater than 50 s are chosen as loop closure samples. We also evaluate the *LCR-Net* trained on our dataset splittings as described in Section VI-A.

Metrics: In line with [5], we employ three metrics to evaluate the registration performance at loop closure as follows:

1) relative translation error (RTE), which measures the Euclidean distance between estimated and ground-truth translation vectors;
2) relative yaw error (RYE), which is the average difference between estimated and ground-truth yaw angle; and
3) registration recall (RR), representing the fraction of scan pairs with RYE and RTE below certain thresholds, e.g., 5° and 2 m.

Results: The baseline methods in this experiment include advanced traditional registration methods, such as ICP [8] and RANSAC [29] with fast persistent feature histograms (FPFH) features [32]. Besides traditional methods, SOTA deep learning approaches are also included, such as RPMNet [72], FCGF [69], DGR [71], Predator [70], CofiNet [39], GeoTransformer [40], RDMNet [41] and LCDNet [5]. Furthermore, we also include results from advanced place recognition methods that can output yaw angles, including Scan Context [9], LiDAR-Iris [24], and OverlapNet [25]. For the hand-crafted method, we use the results reported in [5]. For all deep neural network (DNN)-based approaches, we use its official implementation along with open-source models trained on KITTI odometry datasets. We also report the results of GeoTransformer and

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE II
POINT CLOUD REGISTRATION RESULTS ON CLOSED LOOP OF KITTI ODOMETRY DATASET

		Closed loop with distance below 4 m					
		Seq. 00			Seq. 08		
		RR(%)	RTE(m)[succ./all]	RYE(°)[succ./all]	RR	RTE(m)[succ./all]	RYE(°)[succ./all]
RANSAC-based	RANSAC [29]	33.95	0.98/2.75	1.37/12.01	15.61	1.33/4.57	1.79/37.31
	FCGF [69]	47.31	1.05/2.07	0.60/1.88	27.80	1.28/2.42	2.38/159.48
	Predator [70]	98.93	0.07/0.13	0.11/0.45	99.70	0.10/0.11	0.34/0.53
	CofiNet [39]	**100**	0.07/0.07	0.10/0.10	**100**	0.10/0.10	**0.33/0.33**
	Geotransformer [40]	98.11	0.09/0.28	0.11/0.13	99.12	0.11/0.18	0.35/1.76
	RDMNet [41]	98.36	0.07/0.27	0.11/0.28	99.80	0.10/0.14	0.34/0.49
	LCDNet [5]	**100**	0.11/0.11	0.12/0.12	**100**	0.15/0.15	0.34/0.34
	LCR-Net†	**100**	**0.04/0.04**	**0.09/0.09**	**100**	**0.08/0.08**	**0.33/0.33**
	LCR-Net	100	0.04/0.04	0.09/0.09	100	0.08/0.08	0.33/0.33
RANSAC-free	Scan Context* [9]	97.66	-/-	1.34/1.92	98.21	-/-	1.71/3.11
	LiDAR-Iris* [24]	98.83	-/-	0.65/1.69	99.29	-/-	0.93/1.84
	OverlapNet* [25]	83.86	-/-	1.28/3.89	0.10	-/-	2.03/65.45
	ICP [8]	35.57	0.97/2.08	1.36/8.98	0	-/2.43	-/160.46
	DGR [71]	95.11	0.57/0.74	0.41/2.70	2.42	1.13/7.72	3.13/135.53
	RPMNet [72]	47.31	1.05/2.07	0.60/1.88	27.80	1.28/2.42	1.77/13.13
	LCDNet (fast) [5]	93.03	0.65/0.77	0.86/1.07	60.71	1.02/1.62	1.65/3.13
	Geotransformer (LGR) [40]	96.72	0.10/0.41	0.12/0.99	97.06	0.16/0.35	0.46/2.85
	RDMNet (LGR) [41]	97.57	0.06/0.37	0.12/0.64	99.36	0.10/0.22	0.34/0.59
	LCR-Net† (LGR)	**100**	**0.04/0.04**	**0.09/0.09**	**100**	**0.08/0.08**	**0.33/0.33**
	LCR-Net (LGR)	100	0.04/0.04	0.09/0.09	100	0.07/0.07	0.33/0.33
	LCDNet+ICP [5]	100	0.04/0.04	0.09/0.09	100	0.09/0.09	0.33/0.33
		Closed loop with overlap beyond 0.3					
		Seq. 00			Seq. 08		
		RR(%)	RTE(m)[succ./all]	RYE(°)[succ./all]	RR	RTE(m)[succ./all]	RYE(°)[succ./all]
RANSAC-based	Predator [70]	80.52	0.10/2.13	0.14/13.29	81.98	0.15/1.93	0.37/23.05
	CofiNet [39]	98.21	0.09/0.29	0.13/1.13	99.05	0.13/0.24	0.35/1.49
	Geotransformer [40]	82.07	0.10/3.04	0.13/4.53	92.69	0.15/0.97	0.41/10.29
	RDMNet [41]	71.36	0.13/4.53	0.23/7.81	92.94	0.16/1.37	0.46/6.09
	LCDNet [5]	94.86	0.23/0.89	0.24/3.93	96.87	0.31/0.62	0.51/1.74
	LCR-Net†	**100**	**0.05/0.05**	**0.10/0.10**	**100**	**0.08/0.08**	**0.34/0.34**
	LCR-Net	100	0.05/0.05	0.10/0.10	100	0.08/0.08	0.34/0.34
RANSAC-free	LCDNet (fast) [5]	66.82	0.56/2.02	0.68/6.12	47.29	0.88/3.00	1.06/5.28
	Geotransformer (LGR) [40]	80.43	0.11/3.33	0.16/7.91	87.85	0.20/1.47	0.56/12.78
	RDMNet (LGR) [41]	70.19	0.13/5.06	0.23/11.14	91.62	0.15/1.98	0.48/7.16
	LCR-Net† (LGR)	**100**	**0.04/0.04**	**0.10/0.10**	**100**	**0.08/0.08**	**0.33/0.33**
	LCR-Net (LGR)	100	0.04/0.04	0.10/0.10	100	0.08/0.08	0.34/0.34
	LCDNet+ICP [5]	94.98	0.11/0.77	0.14/3.84	96.98	0.18/0.48	0.43/1.68

Superscript * means the approaches only estimate yaw angle. LCR-Net† is trained on the same sets with baselines and LCR-Net is trained on our sets.

RDMNet using LGR. LCDNet also provides a fast version using weighted SVD. We denote it as LCDNet (fast) and report the results.

Table II shows the results on KITTI odometry sequences 00 and 08. As can be seen, our LCR-Net achieves the best performance across both sequences. LCR-Net is superior in pose estimation, benefiting from dense, reliable matching. We highlight that its pose estimation accuracy exceeds baselines by a large margin and is comparable with LCDNet refined by ICP. As illustrated, the RR for several baselines has reached saturation, reaching 100% in the dataset of the closed loop distance below 4 m. To better showcase the superiority of our approach, we have constructed a new test set comprising point cloud pairs with an overlap ratio exceeding 0.3. The overlap ratio is computed following [25]. These test sets are more challenging, including point cloud pairs with a distance of up to 15 m. We present the results of several competing baselines in former tests. As can be seen, our LCR-Net further amplifies its advantages, being the only method which maintains 100% RR and high pose estimation accuracy, whereas others have all declined.

We highlight that the pose estimation accuracy of LCR-Net surpasses even LCDNet refined by ICP. Especially regarding translation estimation, LCR-Net reduces the error by 64% on sequence 00 and by 16% on sequence 08. This result is encouraging, as current global registration methods typically exhibit inferior registration accuracy compared with geometry-based local registration approaches based on a fine initial guess. As a result, they are commonly employed as initial estimations for methods such as the ICP algorithm in practical applications. In this experiment, LCR-Net demonstrates a noteworthy advancement in registration accuracy compared with the SOTA global registration methods refined by ICP. This significant outcome profoundly underscores the practical advantage of our proposed method in real-world applications.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE III
POINT CLOUD REGISTRATION RESULTS ON CLOSED LOOP OF KITTI360

		Seq. 02			Seq. 09		
		RR(%)	RTE(m)[succ./all]	RYE(°)[succ./all]	RR	RTE(m)[succ./all]	RYE(°)[succ./all]
RANSAC-based	RANSAC [29]	24.78	1.24/3.67	1.83/32.22	29.69	1.12/3.14	1.48/23.42
	FCGF [69]	12.43	0.56/11.32	0.62/146.77	62.40	0.47/5.06	0.44/60.64
	Predator [70]	97.56	0.22/0.34	0.28/1.19	98.44	0.14/0.22	0.18/0.66
	CofiNet [39]	98.56	0.25/0.30	0.30/0.35	**100**	0.15/0.15	0.19/0.19
	Geotransformer [40]	92.32	0.26/0.84	0.49/9.71	96.24	0.16/0.50	0.25/1.84
	RDMNet [41]	94.98	0.23/0.66	0.38/0.97	83.27	0.19/2.24	0.31/3.04
	LCDNet [5]	98.62	0.28/0.32	0.32/0.35	**100**	0.18/0.18	0.20/0.20
	LCR-Net†	**98.63**	**0.21/0.26**	**0.27/0.30**	**100**	**0.11/0.11**	**0.16/0.16**
	LCR-Net	98.64	0.21/0.26	0.27/0.30	100	0.11/0.11	0.16/0.16
RANSAC-free	Scan Context* [9]	92.31	-/-	1.60/5.49	95.29	-/-	1.40/6.80
	LiDAR-Iris* [24]	96.54	-/-	1.71/3.44	86.26	-/-	1.51/7.08
	OverlapNet* [25]	11.42	-/-	1.79/76.74	54.33	-/-	1.38/33.62
	ICP [8]	4.19	1.10/2.26	1.74/149.76	21.24	1.06/2.22	1.34/66.34
	DGR [71]	14.26	0.66/6.68	0.72/126.81	63.51	0.50/3.14	0.38/53.83
	RPMNet [72]	37.99	1.18/2.26	1.30/5.97	41.42	1.13/2.21	1.02/3.95
	LCDNet (fast) [5]	82.92	0.84/1.10	1.28/1.67	89.49	0.76/0.94	0.99/1.19
	Geotransformer (LGR) [40]	91.24	0.25/1.01	0.61/7.90	94.23	0.17/0.70	0.29/2.70
	RDMNet (LGR) [41]	94.44	0.19/0.72	0.36/1.43	80.36	0.18/2.82	0.31/5.64
	LCR-Net† (LGR)	**98.63**	**0.16/0.21**	**0.24/0.27**	**100**	**0.10/0.10**	**0.14/0.14**
	LCR-Net (LGR)	98.59	0.16/0.21	0.23/0.26	100	0.09/0.09	0.14/0.14
	LCDNet+ICP [5]	98.51	0.20/0.25	0.24/0.27	100	0.10/0.10	0.15/0.15

Superscript * means the approaches only estimate yaw angle. LCR-Net† is trained on the same sets with baselines and LCR-Net is trained on our sets.

Table III presents the results on the KITTI-360 datasets. As can be seen, the advantages of our method are still maintained. Two baselines, i.e., CofiNet and LCDNet, demonstrate comparable RR to LCR-Net. However, both methods fall short compared with LCR-Net regarding registration accuracy. In addition, both Cofinet and LCDNet require RANSAC for pose estimation, and LCDNet requires additional ground point filtering.

From the tests, we have observed that RANSAC-based methods generally outperform RANSAC-free methods. However, RANSAC is computationally intensive and can account for more than half of the entire registration process, as evident from our runtime experiments as in Section VI-E. Nevertheless, our proposed method attains the best performance without relying on RANSAC. By employing a fast solver, LGR, LCR-Net attains comparable registration performance to that of RANSAC and even surpasses it in terms of translation estimation accuracy. This characteristic offers a significant advantage for integrating our approach within real-time systems.

In sum, our LCR-Net achieves SOTA performance in closed-loop point cloud registration. It exhibits significant advantages over the baseline methods, offering i) the highest level of registration robustness; ii) Exceptional registration accuracy compared with baseline methods, surpassing even the results obtained by baseline with ICP refinement; and iii) the ability to achieve best performance without relying on RANSAC.

D. Continuous Relocalization Performance

To evaluate the relocalization performance, we use LiDAR pairs at most 10 m apart as samples, which may cause odometry degeneration. We test our approach on KITTI odometry sequences 08–10, KITTI-360, Apollo-SouthBay, Ford Campus, and Mulran datasets. For a fair comparison, we follow [40], [70], and [73] and train the model *LCR-Net*$^{\diamond}$ on KITTI odometry sequences 00–05 and validate it on KITTI odometry sequence 06 and 07. We also evaluate the *LCR-Net* trained on our dataset splittings as described in Section VI-A.

Metrics: We use three metrics to evaluate the registration performance as follows: i) RTE, which measures the Euclidean distance between estimated and ground-truth translation vectors, ii) relative rotation error (RRE), which measures the geodesic distance between estimated and ground-truth rotation matrices, and iii) RR, which represents the fraction of scan pairs with RRE and RTE below certain thresholds, e.g., 5° and 2 m.

Results: We compare the results of our method with the recent RANSAC-based SOTA methods: Predator [70], CofiNet [39], NgeNet [73], GeoTransformer [40], and RDMNet [41]. We also compare our method using LGR with SOTA RANSAC-free methods: HRegNet [42], GeoTransformer [40] and RDMNet [41]. The results are shown in Table IV.

As can be seen, when using RANSAC, our LCR-Net outperforms all the baselines on all the datasets. We highlight the results on the Mulran dataset, where LCR-Net shows remarkable performance by boosting baselines with a large margin for all three metrics, i.e., increasing RR by 10%, reducing rotation error by 43%, and reducing translation error by 20%. The Mulran dataset poses significant challenges for generalization, as it loses approximately 70° field of view (FOV). Besides, when using LGR, LCR-Net attains remarkable results for translation estimation and achieves an average reduction of 35% in translation error compared with the best results attained through RANSAC-based methods across all datasets.

We conduct a qualitative comparison of registration using LCR-Net and LCDNet shown in Fig. 6. We also present ICP alignment by using the LCDNet prediction as an initial guess to demonstrate the accuracy of LCR-Net. The first two columns depict the successful cases of both methods. Our LCR-Net achieves better alignment than LCDNet and LCDNet refined by

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

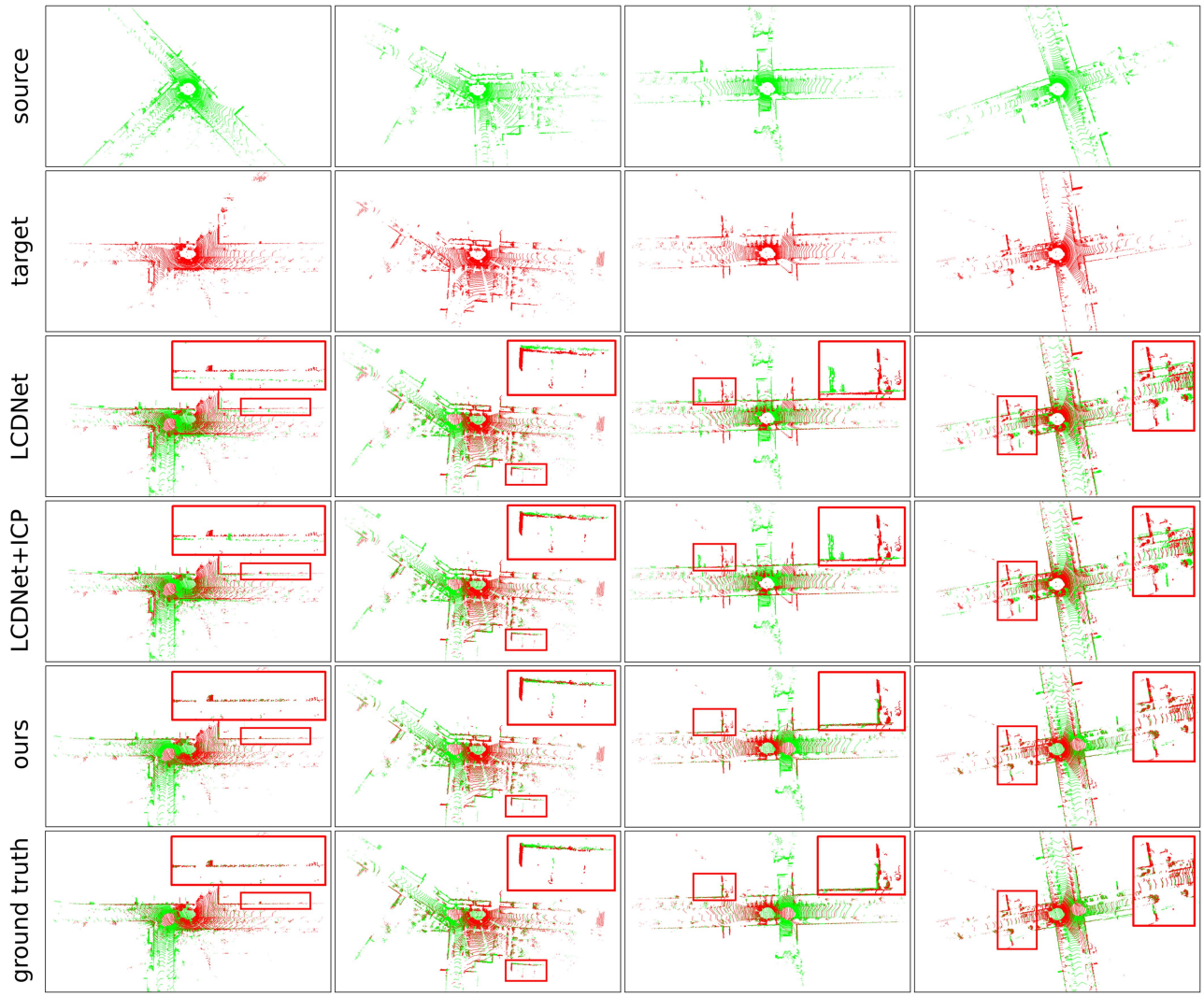

Fig. 6. Qualitative comparison of registration using LCDNet as an initial guess and refined with ICP, and LCR-Net. The first two columns depict the successful cases of both methods, where LCR-Net achieves better alignment. The last two columns show the failure cases of LCDNet, yet LCR-Net still perfectly aligns the two point clouds.

ICP. The last two columns show the failure cases of LCDNet, whereas LCR-Net keeps achieving good alignment.

To provide more insights into the proposed LCR-Net, we visualize the correspondent points founded by LCR-Net as in Fig. 7. The points are aligned using LCR-Net based on LGR. We observe that there are five following characteristics of the regions that the LCR-Net focuses on:

1) it effectively captures overlapping regions of two point clouds;
2) it finds matches over all overlapping regions of two point clouds rather than being limited to a relatively small range, allowing for a wider baseline important for accurate registration;
3) it neglects the majority of ground points;
4) it focuses more on isolated landmarks such as tree trunks, signage, etc.; and
5) it prioritizes important geometric structures such as corners and sloping surfaces.

E. Study on Runtime

We report the runtime of LCR-Net compared with existing SOTA baselines on a system with an Intel i9-10920X CPU and an NVIDIA GTX 3090 GPU. All the methods are evaluated on KITTI sequence 00 with the official implementation. We present the runtime of point cloud registration in Table V. The descriptor extraction time also includes the preprocessing required by the respective method. Pairwise Reg. refers to pair registration using RANSAC as default. As depicted, our LCR-Net using LGR is the fastest method among DNN-based approaches. More commendably, it is also the most robust and accurate one, as shown in Section VI-D and Section VI-C.

In Table V(b), the runtime of loop detection is evaluated. We emphasize that LCR-Net exhibits significantly faster descriptor extraction in candidate retrieval than in point cloud registration. This efficiency is achieved due to our framework, which only activates the KPEncoder and the global description head

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

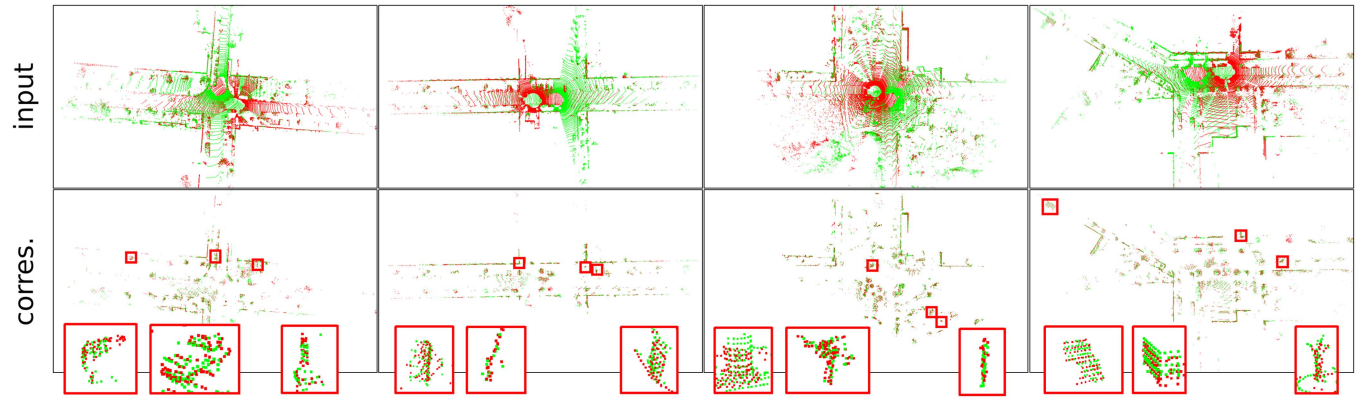

Fig. 7. We visualize the correspondent points found by LCR-Net. For better visualization, we align the input point clouds and the founded correspondent points using LCR-Net based on LGR. For each cases, we zoom in on three representative regions for closer examination. Our LCR-Net effectively finds correspondences covering overall overlapping region and focuses on geometrically significant region.

TABLE IV
CONTINUOUS REGISTRATION RESULTS ON MULTIPLE DATASETS

		KITTI	KITTI-360	Apollo	Ford	Mulran
		RR (%)				
RANSAC-based	Predator [70]	**99.82**	99.50	99.27	99.32	53.02
	CofiNet [39]	**99.82**	99.62	**100**	99.32	80.79
	NgeNet [73]	**99.82**	**99.94**	**100**	**100**	82.96
	Geotransformer [40]	**99.82**	99.86	**100**	**100**	75.68
	RDMNet [41]	**99.82**	99.89	**100**	**100**	87.09
	LCR-Net$^\diamond$	**99.82**	**99.94**	**100**	**100**	**97.89**
	LCR-Net	99.82	99.94	100	100	98.22
RANSAC-free	HRegNet [42]	96.76	20.39	9.39	90.48	-
	Geotransformer (LGR) [40]	**99.82**	99.86	**100**	98.64	72.91
	RDMNet (LGR) [41]	**99.82**	99.90	**100**	**100**	83.68
	LCR-Net$^\diamond$ (LGR)	**99.82**	**99.94**	**100**	**100**	**96.76**
	LCR-Net (LGR)	99.82	99.94	100	100	97.53
		RRE (°)				
RANSAC-based	Predator [70]	0.25	0.29	0.21	0.38	1.03
	CofiNet [39]	0.37	0.44	0.18	0.37	0.52
	NgeNet [73]	0.26	0.30	0.18	0.26	0.35
	Geotransformer [40]	0.22	0.28	0.28	0.28	0.30
	RDMNet [41]	0.18	0.25	**0.10**	0.21	0.45
	LCR-Net$^\diamond$	**0.17**	**0.22**	**0.10**	**0.18**	**0.17**
	LCR-Net	0.19	0.24	0.09	0.16	0.17
RANSAC-free	HRegNet [42]	1.04	2.18	2.14	0.91	-
	Geotransformer (LGR) [40]	0.31	0.36	0.29	0.50	0.49
	RDMNet (LGR) [41]	**0.27**	**0.35**	**0.29**	**0.39**	0.48
	LCR-Net$^\diamond$ (LGR)	0.33	0.36	0.30	0.42	**0.45**
	LCR-Net (LGR)	0.35	0.36	0.29	0.34	0.43
		RTE (cm)				
RANSAC-based	Predator [70]	5.8	7.2	7.8	11.7	30.04
	CofiNet [39]	8.2	10.1	6.7	11.6	17.3
	NgeNet [73]	6.1	7.5	5.9	7.9	9.2
	Geotransformer [40]	6.7	8.1	6.1	10.6	12.0
	RDMNet [41]	5.3	7.0	4.6	7.6	14.4
	LCR-Net$^\diamond$	**3.9**	**5.7**	**3.6**	**7.1**	**7.5**
	LCR-Net	3.9	5.8	3.4	6.6	7.4
RANSAC-free	HRegNet [42]	6.7	112.8	116.7	16.5	-
	Geotransformer (LGR) [40]	5.5	6.9	5.1	12.2	9.7
	RDMNet (LGR) [41]	3.9	5.9	3.5	**6.1**	9.3
	LCR-Net$^\diamond$ (LGR)	**2.9**	**5.0**	**2.5**	**6.1**	**6.2**
	LCR-Net (LGR)	2.8	5.4	2.5	6.5	6.2

LCR-Net$^\diamond$ is trained on the same sets with baselines and LCR-Net is trained on our sets.

TABLE V
COMPARISON OF INFERENCE TIME

(a) Inference time for point cloud registration.

	Model	Descriptor Extraction (ms)	Pairwise Reg. (ms)	Total (ms)
Hand	RANSAC	135	156	291
	FGR	135	246	381
DNN-based	Predator	173	244	417
	CofiNet	377	257	634
	LCDNet	235	431	666
	LCDNet+ICP	235	566	801
	LCR-Net	293	390	683
	LCR-Net (LGR)	293	13	306

(b) Inference time for candidate retrieval.

	Model	Descriptor Extraction (ms)	Pairwise Comp. (ms)	Map querying (ms)
Hand	Scan Context	1	0.09	6
	LiDAR-Iris	10	9	42037
DNN	OverlapNet	15	6	26634
	LCDNet	182	0.01	5
	LCR-Net	50	0.01	5

during inference, eliminating the need for complex point-level descriptor generation. Regarding map querying, we report the runtime for searching a descriptor in all its previous scans. For the pairwise comparison, both LiDAR-Iris and OverlapNet employ an ad-hoc comparison function that is hard to optimize in runtime, whereas LCR-Net and LCDNet use the Euclidean distance. Therefore, we can use the FAISS library [62] for fast searching in LCR-Net and LCDNet. We report the querying time of Scan Context using the ring key descriptor for fast searching. As depicted, our LCR-Net demonstrates exceptional efficiency in loop retrieval with the support of FAISS.

In summary, LCR-Net exhibits high efficiency in both registration and candidate retrieval tasks, rendering it well-suited for online SLAM applications.

F. Evaluation of Complete SLAM System

In the previous experiments, we have demonstrated the superior performance and high efficiency of the proposed LCR-Net

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

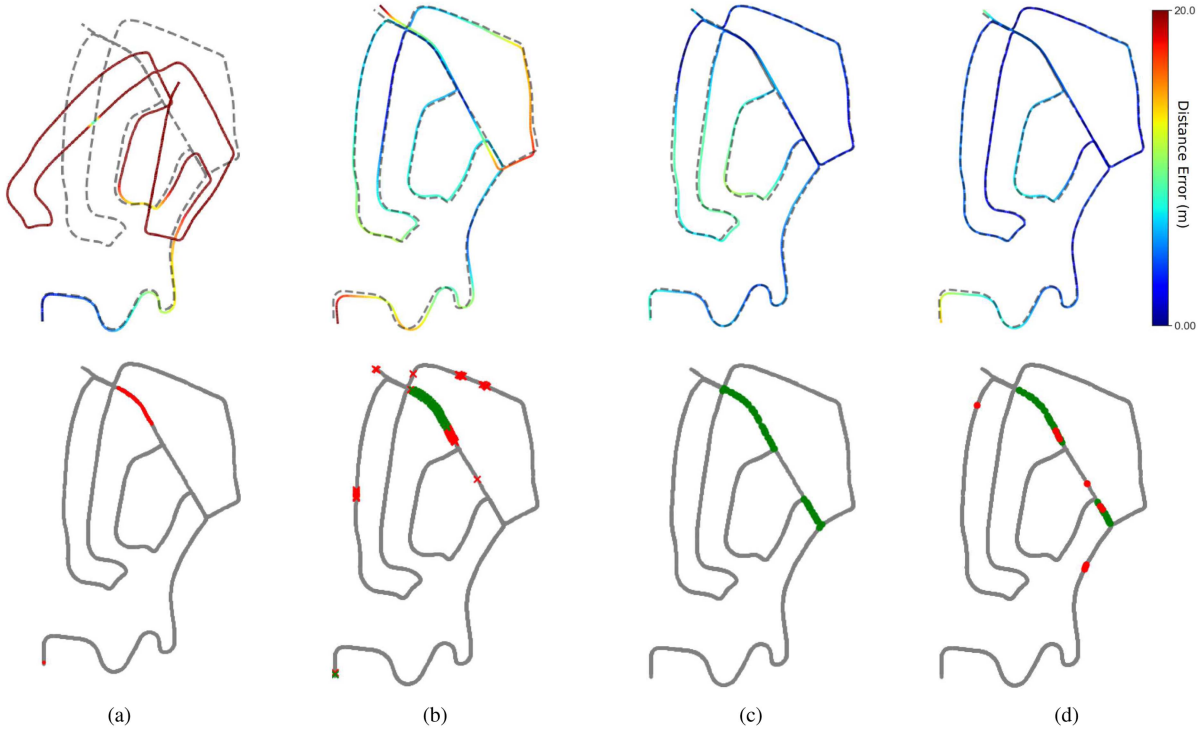

Fig. 8. Performance of A-LOAM with integrating LCR-Net compared with integrating Scan Context on KITTI sequence 02. We present the results. (a) Original A-LOAM (top) and degenerated keyframes colored in red (bottom). (b) A-LOAM with relocalization using LCR-Net (top) and the detected degenerated keyframes (bottom, green crosses × are true positive and red crosses × are false positive detections). (c) A-LOAM with relocalization and loop closing using LCR-Net (top) and the detected loop closures (bottom, green dots • are true positive detections). (d) A-LOAM with relocalization using LCR-Net and loop closing using Scan Context and ICP (top) and the detected loop closures (bottom, green dots • are true positive detections and red dots • are false positive).

in specific subtasks of loop closing and relocalization. We have also verified that the model trained on our dataset splittings, *LCR-Net*, can achieve comparable SOTA performance with the models specifically trained for each task, *LCR-Net*† and *LCR-Net*$^{\diamond}$. In this section, we will evaluate the performance of our network in solving loop closing and relocalization challenges when integrated with a SLAM system. We use the SLAM system with A-LOAM as the front-end odometry. We demonstrate the gradual improvement of SLAM results by integrating our *LCR-Net* using the LGR estimator for relocalization and loop closing, as depicted in Fig. 8. The relocalization operates at a frequency of 2.5 Hz, whereas loop closing is configured to operate at a frequency of 1 Hz.

The top row of Fig. 8(a) shows the original results of A-LOAM. As seen, A-LOAM suffers severe degeneration and fails to provide satisfactory results. We determine a keyframe's ground truth degeneration flag by comparing the relative pose estimation error to a predefined threshold. When the pose error exceeds 1 m, the current keyframe is considered degenerated. The bottom row of Fig. 8(a) highlights the degenerated keyframes on the ground-truth trajectory in red.

The bottom row of Fig. 8(b) shows the degenerated keyframes detected by our approach. Our approach effectively detects all the degenerations. Although the introduction of numerous false positives results in a slight increase in the number of calls made to the relocalization module, it effectively prevents system degradation, making it an acceptable tradeoff for practical applications. The SLAM results are significantly improved by enabling the relocalization module, as in the top row of Fig. 8(b). The trajectory is colored with the distance error between our estimations and ground-truth poses.

The bottom row of Fig. 8(c) shows the loop detected and closed by our approach. As in the top row of Fig. 8(c), by enabling both relocalization and loop closing module, the results are further refined compared with Fig. 8(b).

As the code of LCDNet with SLAM is currently unavailable, we compare the results with the most commonly employed loop closing approach, which exploits Scan Context for loop detection and ICP for registration, as in Fig. 8(d). The bottom row of Fig. 8(d) shows the loop detected and closed by Scan Context. As depicted, it introduces several false positives, whereas our approach successfully avoids them. Compared with our approach, using Scan Context and ICP produces a significant deviation at the starting position (bottom-left), resulting in a higher absolute position error (APE) of 4.9 m, which is 4.2 m for our approach.

We assess different systems' APE on KITTI odometry sequences 00 and 08 as well as on NCLT [74] sequence 2013-01-10 acquired at Michigan University using a Velodyne HDL32. This dataset presents a notable complexity with the high-frequency motion profile, stemming from the inherent instability of the platform-a two-wheeled Segway, which poses a significant challenge for ICP-based odometry methods [47], [75]. The results of our system and the system employing Scan Context and ICP (SC+ICP) as the loop closing pipeline are presented in Fig. 9. By leveraging the flexible structure of the

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
ABLATION STUDIES OF INDIVIDUAL MODULES

(a) **Rotary embedding in 3D-RoFormer++**. Linear embedding is effective.

case	Mulran			KITTI-loop		
	RR(%)	RRE(°)	RTE(cm)	RR(%)	RRE(°)	RTE(cm)
3D-RoFormer	94.97	0.18	8.1	99.86	0.29	5.9
3D-RoFormer++	**98.22**	**0.17**	**7.4**	**100**	**0.21**	**4.3**
3D-RoFormer++*	94.29	0.18	7.5	**100**	0.23	**4.3**

(b) VoteEncoder. The full VoteEncoder performs the best.

case	Mulran			KITTI-loop		
	RR(%)	RRE(°)	RTE(cm)	RR(%)	RRE(°)	RTE(cm)
full	**98.22**	**0.17**	**7.4**	**100**	**0.21**	**4.3**
w/o KPConv	93.40	0.18	8.0	99.97	0.26	5.0
w/o central prediction	92.80	0.19	8.0	99.99	0.24	4.5

(c) **Overall structure**. Both VoteEncoder and 3D-RoFormer++ contribute to enhance the registration robustness and accuracy.

case	Registration						Candidate retrieval							
	Mulran			KITTI-loop			KITTI				Campus			
	RR(%)	RRE(°)	RTE(cm)	RR(%)	RRE(°)	RTE (cm)	AUC	F1max	Recall@1	Recall@1%	AUC	F1max	Recall@1	Recall@1%
full	**98.22**	**0.17**	**7.4**	**100**	**0.21**	**4.3**	0.958	0.922	0.937	**0.993**	0.972	**0.920**	0.932	0.987
w/o VoteEncoder	85.46	0.41	20.1	99.99	0.24	6.6	**0.975**	**0.940**	**0.948**	0.990	**0.973**	0.917	0.927	0.981
w/o 3D-RoFormer++	93.67	0.22	10.1	**100**	0.26	4.8	0.950	0.919	0.920	0.962	0.951	0.871	**0.939**	**0.994**

(d) **Training strategy**. Linear probe ensures the best registration performance while allowing for enhanced candidate retrieval performance.

case	b	p	n	Registration						Candidate retrieval							
				Mulran			KITTI-loop			KITTI				Campus			
				RR(%)	RRE(°)	RTE(cm)	RR(%)	RRE(°)	RTE (cm)	AUC	F1max	Recall@1	Recall@1%	AUC	F1max	Recall@1	Recall@1%
linear probe	6	6	6	**98.22**	**0.17**	**7.4**	**100**	**0.21**	**4.3**	**0.958**	**0.922**	**0.937**	**0.993**	**0.972**	**0.920**	**0.932**	**0.987**
fine-tune	1	1	1	90.13	0.18	8.1	**100**	**0.21**	4.5	0.900	0.834	0.911	0.975	0.883	0.818	0.877	0.971
end-to-end	1	1	1	85.70	0.21	9.6	**100**	0.22	4.7	0.868	0.808	0.872	0.966	0.834	0.782	0.871	0.979

Default structure are marked in gray and the best results are highlighted in bold.

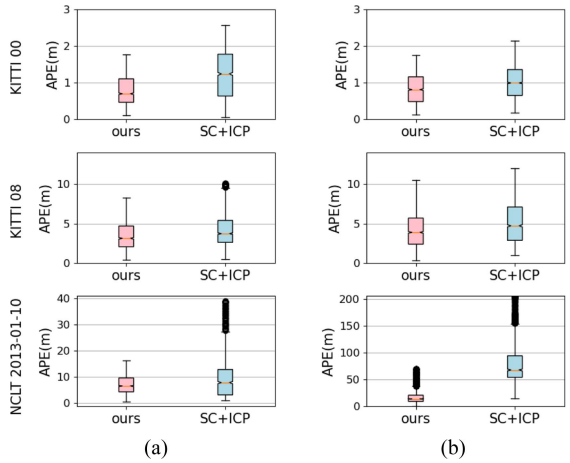

Fig. 9. Comparison of SLAM systems with different front-end odometry approaches on different sequences. (a) Front end: A-LOAM. (b) Front end: F-LOAM.

PGO framework, we also substitute the front-end odometry with F-LOAM [76], thereby demonstrating the capability of our method to integrate diverse odometry systems. As depicted in Fig. 9, regardless of the chosen odometry front-end, our approach consistently yields more precise localization results than systems utilizing Scan Context and ICP. Moreover, our approach exhibits smaller errors and enhanced stability. Our system demonstrates a notable superiority, particularly evident in handling the challenging NCLT dataset. When employing F-LOAM as the front-end, the system suffers from severe degeneracy and subsequent localization failure. Our method expertly overcomes such system failures and considerably reduces APEs. In conclusion, our experiments demonstrate the significance of our method, particularly in highly challenging scenarios where it serves as a crucial safeguard against critical degeneracy.

G. Ablation Study

We conduct ablation studies to better understand the effectiveness of each module in our proposed LCR-Net, as in Table VI. Table VI(a) and (b) study specific designs of 3D-RoFormer++ and VoteEncoder in continuous registration tasks using Mulran dataset and closed-loop registration task using KITTI dataset (loop with distance below 4 m). Table VI(c) and Table VI(d) study overall structure and training strategy in registration task and candidate retrieval task using KITTI and Ford Campus datasets. We use the model trained on our dataset splittings, *LCR-Net*, as the default model. The registration performance is evaluated using the RANSAC estimator.

Rotary embedding: Table VI(a) studies the influence of different rotary position embedding used in 3D-RoFormer++, where i) 3D-RoFormer uses nonlinear embedding as in (1); ii) 3D-RoFormer++ uses the linear embedding as in (7); and iii) 3D-RoFormer++* refers to the 3D-RoFormer++ trained without penalty loss. Using a linear rotary position embedding improves registration robustness and accuracy. The reason could be attributed to the network's ability to consistently represent relative positions through a linear embedding, thus enhancing the registration performance. Using the penalty loss can further help the position embedding learning.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

VoteEncoder: Table VI(b) studies the designs of VoteEncoder. Leveraging the central prediction algorithm, VoteEncoder acquires a more robust estimation for the center of the interested region, thus enhancing registration accuracy. The incorporation of feature aggregation capability via KPConv improves both robustness and accuracy.

Overall structure: Table VI(c) studies the impact of our VoteEncoder and 3D-RoFormer++. As can be seen, VoteEncoder significantly enhances the registration performance. The underlying factors contributing to this improvement are examined in Section VI-H. We study the 3D-RoFormer++ by replacing it with a vanilla Transformer using an absolute position encoder [38]. As depicted, 3D-RoFormer++ exhibits superior advantages in registration compared with the vanilla Transformer.

Though 3D-RoFormer++ does not directly contributes to global descriptor generation, it still impacts performance. As illustrated, 3D-RoFormer++ is beneficial in enhancing the generalization of the global descriptor, which suggests that 3D-RoFormer++ is more effective in improving the backbone's feature learning than vanilla Transformer. On the other hand, using the VoteEncoder does not have a significant impact. The performance tradeoff is justified by the substantial improvement in registration brought by VoteEncoder.

Training strategy: In Table VI(d), we study the influence of training strategy. We compare the training strategy used in this article, linear probe, with two alternative strategies. The first strategy is fine-tuning, where we also pretrain the model in the registration task but subsequently fine-tune the model in the candidate retrieval task. The second strategy is end-to-end, which trains the whole network from scratch. With the principle of maximizing computational resource utilization, we have set the batch size (b) to 1, with $N_p = 1$ positive scan (p) and $N_n = 1$ negative scan (n) for these two strategies. As can be seen, we are constrained to much smaller batch sizes and fewer positive/negative scans in these two strategies than in the linear probe strategy, which largely constrains the candidate retrieval performance. Note that all the strategies use the same batch size as 1 when trained on registration, yet the linear probe still achieves the best registration performance. This can be attributed to the linear probe strategy not facing the tradeoff between global descriptor capacity and registration capability during training.

H. Insights on How VoteEncoder Contributes to Registration

In our ablation study, we observe significant registration performance enhancement brought about by our VoteEncoder. To understand how VoteEncoder contributes to registration, we further conduct insight experiments on six additional metrics as follows:

1) IR, the ratio of correct correspondences with residuals below a certain threshold, e.g., 0.6 m, after applying the ground-truth transformation;
2) match recall (MR), the ratio of true point matches found among all the true matches;
3) hit ratio (HR), the fraction of true points that have matches among all points that have matches;
4) patch inlier ratio (PIR);
5) patch match recall (PMR);
6) patch hit ratio (PHR) are with the same meaning with IR, MR, and HR, respectively, but at sparse matching level.

TABLE VII
COMPREHENSIVE EVALUATION OF VOTEENCODER ON CONTINUOUS REGISTRATION TASK USING THE KITTI DATASET

Model	Direct metrics			Indirect metrics					
	RR(%)	RRE(°)	RTE(cm)	IR	MR	HR	PIR	PMR	PHR
full	**99.82**	**0.17**	**3.9**	78.2	**9.2**	**69.2**	80.9	**62.8**	**86.3**
w/o VoteEncoder	**99.82**	0.22	6.4	**84.5**	3.7	35.1	**97.0**	15.7	43.2
5KPEncoder	**99.82**	0.20	4.6	81.2	6.3	51.0	88.6	39.4	77.5

The best results are highlighted in bold.

The true patch match represents that patch pairs have actual overlap. MR and PMR evaluate the coverage ratio of found matches among all true matches, whereas HR and PHR better assess the match distribution over the overlapping region since a point/patch may possesses multiple matches.

We present the results of LCR-Net and LCR-Net without VoteEncoder in Table VII. To evaluate the efficacy of the VoteEncoder in contrast to an alternative downsampling and aggregation approach, we also present a variant, denoted as *5KPEncoder*, that replaces VoteEncoder with another KPEncoder. As can be seen, using VoteEncoder has shown decreased IR and PIR but notably increased PMR and HR. The rationale behind this can be explained as follows: the VoteEncoder can be seen as a downsampler that filters redundant points and aggregates them into new keypoints. After filtering massive points with similar neighbors, it becomes more difficult to find true matches in sparser candidates (as observed by decreased PIR). However, this also allows LCR-Net to focus on searching different geometrically salient regions over the point cloud rather than repeatedly sampling the same salient region (as observed by increased PMR and HR). Moreover, this also leads to a more reasonable point patch grouping, making it easier to match an object as a whole and simplifying the subsequent dense matching process (see the decrease in PIR by 17.9%, whereas IR only decreases by 6.3%). As depicted, the significant improvement in PMR and HR brought by VoteEncoder, allowing for wider baselines and more dimensional constraints in registration, greatly reduces registration errors (see the decrease in RRE by 23% and RTE by 39%). Comparison between *5KPEncoder* and full LCR-Net demonstrates that the VoteEncoder achieves more effective downsampling and better registration than a KPEncoder.

We provide qualitative comparisons of matching points in Fig. 10. As depicted, matching points obtained by full LCR-Net offer enhanced coverage of overlapping regions within the point clouds. We zoom in and examine four specific local areas within both scenes. It is evident that the full LCR-Net effectively groups and matches points of local areas.

Insights: The above results lead us to rethink which indirect metrics are more important in point cloud registration. Apart from the final performance evaluated by direct metrics of RR, RRE, and RTE, most prior works [33], [34], [35], [36], [37], [39], [40], [41], [69], [70], [73] focus only on the keypoint salience evaluated by indirect metric IR, also referred to as precision. However, higher IR does not guarantee higher robustness or

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

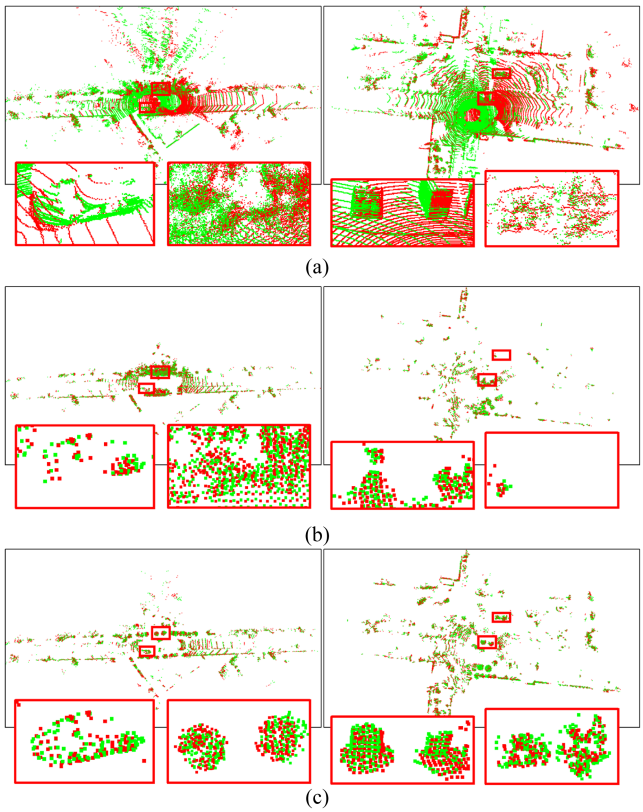

Fig. 10. We visualize the matching points identified by LCR-Net to demonstrate the focus of our LCR-Net. The matching points obtained by LCR-Net after ablating VoteEncoder is also visualized for comparison. For better visualization, we align the points in two scans using the pose estimated by LCR-Net based on LGR. In each case, we zoom in on three representative regions for closer examination. (a) Input point clouds. (b) Matching points found by LCR-Net ablating VoteEncoder. (c) Matching points found by LCR-Net.

accuracy, as observed in Table VII. Conversely, the coverage ratio of found matches among all true matches, as evaluated by metrics MR and HR, should be a crucial factor to consider. These metrics provide insights into the distribution of matching points and can guide the design of new keypoint detection methods by considering this distribution.

Indeed, numerous existing keypoint detection and matching methods [36], [37] have proposed to prevent keypoints from being excessively concentrated in local areas. Early work for LiDAR odometry, LOAM [61] and its variants [76], [77] also propose to extract keypoints uniformly from all LiDAR scan lines. However, they have not provided a clear explanation for the underlying reasons. We are the first to specifically propose this insight and validate it through experimental cases by providing quantitative metrics.

VII. Conclusion

This article analyzes the challenges and commonalities between loop closing and relocalization tasks and proposes a unified framework to address both. Within this framework, we present LCR-Net, a novel multihead network for candidate retrieving and point cloud registration. LCR-Net incorporates a novel keypoint detection module, offering three types of features to meet different requirements for distinct postprocessing. Two novel submodules, 3D-RoFormer++ and VoteEncoder, are proposed for feature generation. 3D-RoFormer++ learns contextual information between two point clouds and leverages a novel translation-invariant rotary position embedding for geometric information aggregation. VoteEncoder, on the other hand, learns to detect keypoints from distinct objects while maintaining a uniform and nonrepeating distribution. We demonstrate the SOTA performance of our LCR-Net in three subtasks related to loop closing and relocalization. Given the notable improvement in registration performance brought by VoteEncoder, we conduct experiments to analyze the underlying reasons and obtain an important insight that point matching should consider distribution rather than solely focusing on precision. Finally, we integrate our network as the loop closing and relocalization module into a complete SLAM system and verify its capability to address loop closing and relocalization tasks. In this article, we employ spectral analysis of the Jacobian matrix as the criterion for detecting degeneracy. However, this approach introduces numerous false detections, which do not influence the localization results yet bring extra computational costs. In future work, we will explore techniques for more efficient degeneracy detection to enhance the performance of our system.

References

[1] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard, "Place recognition in 3D scans using a combination of bag of words and point feature based relative pose estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 1249–1255.

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

[3] Y. Cui, X. Chen, Y. Zhang, J. Dong, Q. Wu, and F. Zhu, "BoW3D: Bag of words for real-time loop closing in 3D LiDAR SLAM," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2828–2835, May 2023.

[4] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 1470–1477.

[5] D. Cattaneo, M. Vaghi, and A. Valada, "LCDNet: Deep loop closure detection and point cloud registration for LiDAR SLAM," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2074–2093, Aug. 2022.

[6] J. Arce, N. Vodisch, D. Cattaneo, W. Burgard, and A. Valada, "PADLOC: LiDAR-based deep loop closure detection and registration using panoptic attention," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1319–1326, Mar. 2023.

[7] K. Vidanapathirana, P. Moghadam, S. Sridharan, and C. Fookes, "Spectral geometric verification: Re-ranking point cloud retrieval for metric localization," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2494–2501, May 2023.

[8] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

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

[10] J. Knopp, J. Sivic, and T. Pajdla, "Avoiding confusing features in place recognition," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 748–761.

[11] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. 12th Int. Conf. Comput. Vis. Workshops*, 2009, pp. 689–696.

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

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

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

[14] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.
[15] J. Komorowski, "MinkLoc3D: Point cloud based large-scale place recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1790–1799.
[16] J. Komorowski, "Improving point cloud based place recognition with ranking-based loss and large batch training," in *Proc. Int. Conf. Pattern Recognit.*, 2022, pp. 3699–3705.
[17] K. Vidanapathirana, M. Ramezani, P. Moghadam, S. Sridharan, and C. Fookes, "Logg3D-Net: Locally guided global descriptor learning for 3D place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2215–2221.
[18] F. Radenovic, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019.
[19] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[20] L. Hui, H. Yang, M. Cheng, J. Xie, and J. Yang, "Pyramid point cloud transformer for large-scale place recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6098–6107.
[21] Y. Xia et al., "SOE-Net: A self-attention and orientation encoding network for point cloud based place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11348–11357.
[22] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10529–10538.
[23] S. S. Harithas et al., "FinderNet: A data augmentation free canonicalization aided loop detection and closure technique for point clouds in 6-DOF separation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 8384–8393.
[24] Y. Wang, Z. Sun, J. Yang, and H. Kong, "LiDAR IRIS for loop-closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5769–5775.
[25] X. Chen et al., "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot.: Sci. Syst.*, 2020.
[26] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Overlaptransformer: An efficient and yaw-angle-invariant transformer network for lidar-based place recognition," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6958–6965, Jul. 2022.
[27] J. Ma, X. Chen, J. Xu, and G. Xiong, "SEQOT: A spatial–temporal transformer network for place recognition using sequential LiDAR data," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 8225–8234, Aug. 2023.
[28] J. Ma, G. Xiong, J. Xu, and X. Chen, "CVTNet: A cross-view transformer network for LiDAR-Based place recognition in autonomous driving environments," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 4039–4048, Mar. 2024.
[29] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[30] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994.
[31] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot. Sci. Syst.*, 2009, pp. 1–7.
[32] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.
[33] H. Deng, T. Birdal, and S. Ilic, "PPFNet: Global context aware local features for robust 3D point matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 195–205.
[34] H. Deng, T. Birdal, and S. Ilic, "PPF-Foldnet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 602–618.
[35] Z. J. Yew and G. H. Lee, "3DFEAT-NET: Weakly supervised local 3D features for point cloud registration," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 607–623.
[36] J. Li and G. H. Lee, "USIP: Unsupervised stable interest point detection from 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 361–370.
[37] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3D local features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6359–6367.
[38] C. Shi, X. Chen, K. Huang, J. Xiao, H. Lu, and C. Stachniss, "Keypoint matching for point cloud registration using multiplex dynamic graph attention networks," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8221–8228, Oct. 2021.
[39] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 23872–23884.
[40] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11143–11152.
[41] C. Shi, X. Chen, H. Lu, W. Deng, J. Xiao, and B. Dai, "RDMNET: Reliable dense matching based point cloud registration for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 11372–11383, Oct. 2023.
[42] F. Lu et al., "HREGNET: A hierarchical network for large-scale outdoor LiDAR point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16014–16023.
[43] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "OverlapNet: A Siamese network for computing LiDAR scan similarity with applications to loop closing and localization," *Auton. Robots*, vol. 46, pp. 61–81, 2021.
[44] J. Du, R. Wang, and D. Cremers, "DH3D: Deep hierarchical 3D descriptors for robust large-scale 6dof relocalization," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 744–762.
[45] J. Komorowski, M. Wysoczanska, and T. Trzciński, "EGONN: Egocentric neural network for point cloud based 6dof relocalization at the city scale," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 722–729, Apr. 2022.
[46] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 809–816.
[47] W. Xu and F. Zhang, "Fast-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.
[48] H. Zhou, Z. Yao, and M. Lu, "Lidar/UWB fusion based SLAM with anti-degeneration capability," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 820–830, Jan. 2021.
[49] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 1–9.
[50] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa : Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.
[51] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
[52] M. Zins, G. Simon, and M.-O. Berger, "OA-SLAM: Leveraging objects for camera relocalization in visual SLAM," in *Proc. Int. Symp. Mixed Augmented Reality*, 2022, pp. 720–728.
[53] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6411–6420.
[54] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.
[55] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J.-H. Wan, and Y. Guo, "Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18953–18962.
[56] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4938–4947.
[57] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, pp. 876–879, 1964.
[58] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
[59] P. K. Diederik and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 7–9, 2015.
[60] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, pp. 216–235, 2012.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.

[61] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, 2014, pp. 1–9.
[62] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.
[63] 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.*, 2012, pp. 3354–3361.
[64] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023.
[65] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: Towards learning based LiDAR localization for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6389–6398.
[66] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6433–6438.
[67] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognit.*, vol. 113, 2021, Art. no. 107760.
[68] T. Röhling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-D LIDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 736–741.
[69] C. B. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.
[70] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3D point clouds with low overlap," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4267–4276.
[71] C. B. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2514–2523.
[72] Z. J. Yew and G. H. Lee, "RPM-Net: Robust point matching using learned features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11824–11833.
[73] L. Zhu, H. Guan, C. Lin, and R. Han, "Neighborhood-aware geometric encoding network for point cloud registration," 2022, *arXiv:2201.12094*.
[74] N. C.-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and LiDAR dataset," *Int. J. Robot. Res.*, vol. 35, pp. 1023–1035, 2016.
[75] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic LiDAR odometry with loop closure," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5580–5586.
[76] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM: Fast LiDAR odometry and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 4390–4396.
[77] T. Shan and B. Englot, "Lego-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.

Chenghao Shi received the bachelor's of engineering degree in detecting guidance and control technology from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2017, the master's degree in control science and engineering in 2019 from the National University of Defense Technology, Changsa, China, where he is working toward the Ph.D. degree in control science and engineering.

His research interests include localization and mobile robots.

Xieyuanli Chen received the bachelor's degree in electrical engineering and automation from Hunan University, Changsha, China, in 2015, the master's degree in robotics from the National University of Defense Technology (NUDT), Changsha, in 2017, and the Ph.D. degree in engineering from Photogrammetry and Robotics Laboratory, the University of Bonn, Bonn, Germany, in 2022.

In 2023, he joined the College of Intelligence Science and Technology, NUDT, where he is now an Associate Professor.

Junhao Xiao (Senior Member, IEEE) received the B.E. degree in automation from the National University of Defense Technology, Changsha, China, in 2007, and the Ph.D. degree in computer science from the Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Informatics, University of Hamburg, Hamburg, Germany. In 2013, he joined the College of Intelligence Science and Technology, NUDT, where he is currently an Associate Professor since 2017. His research interest focuses on mobile robotics, especially localization, mapping, and path planning.

Bin Dai received the Ph.D. degree in control science and engineering from National University of Defense Technology, Changsha, China, in 1998.

In 2006, he was a Visiting Scholar with the Intelligent Process Control and Robotics Laboratory, Karlsruhe Institute of Technology. He is currently a Professor with the Unmanned Systems Research Center, National Innovation Institution of Defense Technology, Beijing, China. His research interests include computer vision and intelligent vehicles.

Huimin Lu received the B.E. degree in automation, the M.E. and Ph.D. degrees in control science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2003, 2005, and 2010, respectively.

In 2010, he joined the College of Intelligence Science and Technology, NUDT, where he is currently a Professor. His research interests include mobile robotics, mainly robot vision, multirobot coordination, and human-robot interaction.

Authorized licensed use limited to: University of Chinese Academy of SciencesCAS. Downloaded on October 19,2025 at 12:05:26 UTC from IEEE Xplore. Restrictions apply.