2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Unified BEV Model for Joint Learning of 3D Local Features and Overlap Estimation

Lin Li[1,2], Wendong Ding[1], Yongkun Wen[1], Yufei Liang[2], Yong Liu[2,*] and Guowei Wan[1,*]

***Abstract*— Pairwise point cloud registration is a critical task for many applications, which heavily depends on finding correct correspondences from the two point clouds. However, the low overlap between input point clouds causes the registration to fail easily, leading to mistaken overlapping and mismatched correspondences, especially in scenes where non-overlapping regions contain similar structures. In this paper, we present a unified bird's-eye view (BEV) model for jointly learning of 3D local features and overlap estimation to fulfill pairwise registration and loop closure. Feature description is performed by a sparse UNet-like network based on BEV representation, and 3D keypoints are extracted by a detection head for 2D locations, and a regression head for heights. For overlap detection, a cross-attention module is applied for interacting contextual information of input point clouds, followed by a classification head to estimate the overlapping region. We evaluate our unified model extensively on the KITTI dataset and Apollo-SouthBay dataset. The experiments demonstrate that our method significantly outperforms existing methods on overlap estimation, especially in scenes with small overlaps. It also achieves top registration performance on both datasets in terms of translation and rotation errors.**

I. INTRODUCTION

Pairwise point cloud registration aims to align two partially overlapped point clouds, which is a fundamental task in many applications, such as LiDAR SLAM [1], [2], LiDAR-based mapping [3], [4], and localization [5], [6]. Another equally important module in the SLAM system is loop closure, which ensures a globally consistent map. Recent works have made substantial progress in loop closure detection [7]–[10] and point cloud registration [11]–[13]. For loop closure detection, it is common practice to encode the entire point cloud into a global descriptor [7], [8]. The advantage of this encoding method is that it is lightweight and convenient for retrieval. However, due to the lack of information interaction, this encoding is not robust to occlusions and small overlaps. The same problem exists in the field of point cloud registration. Some recent point cloud registration works [12], [14], [15] have begun to focus on small overlapping scenarios. However, most of these works are mainly aimed at indoor scenes. Point cloud registration of outdoor scenes with low overlap is very challenging because the point cloud gets sparser with distance.

Loop closure detection is inherently related to overlap estimation, and the latter can be considered as a similarity metric. Intuitively, pairwise registration is directly affected by the overlap of the two point clouds; e.g., the overlap can be utilized to filter out mismatched correspondences. To go further, in the training stage, the overlap can effectively supervise the contrastive learning of the two input point clouds for feature description and keypoint detection. Therefore, overlap estimation is important for loop closure and pairwise registration of point clouds.

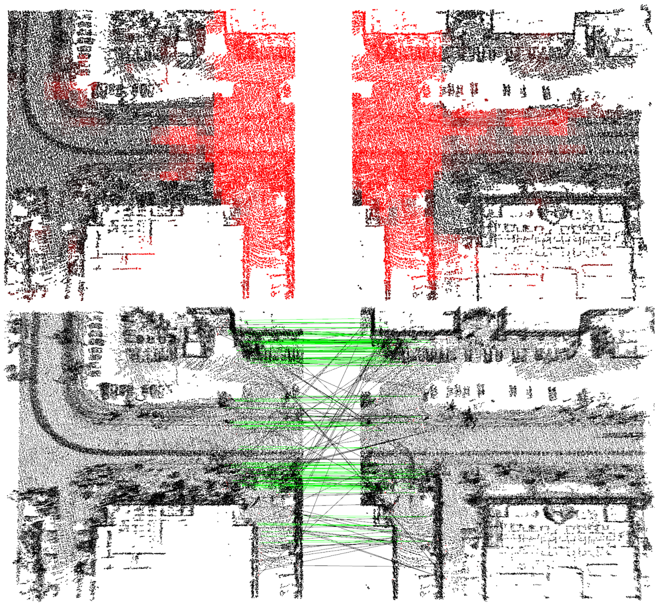

Fig. 1: An illustration of our method. The upper part of the figure shows the detected overlapping regions (in red) in an outdoor scene. The lower part shows the correspondences (inliers in green and outliers in black) found by our method.

In this work, we seek to jointly learn overlap estimation and 3D local features in a unified BEV model. BEV form is a compact and natural representation of point clouds in outdoor scenes, which are usually collected using LiDAR sensors mounted on vehicles. We represent input point clouds as multi-layer BEVs and apply a UNet-like network to extract multi-scale features. We detect keypoints on the 2D BEV representation and extract local descriptors. For 6-DoF registration, we obtain the height of each 2D BEV cell in a regression way. We adopt a cross-attention module to interact with the input point clouds and then perform overlapping region classification on the 2D BEV plane. For pairwise registration, we only detect corresponding keypoints within the overlapping area. For loop closure detection, we take the area of the overlapping region as a measure of their similarity. Fig. 1 is an illustration of our method.

[1] Wendong Ding, Yongkun Wen and Guowei Wan are with Baidu Intelligent Driving Group, Beijing 100094, P. R. China. This work is done when Lin Li is an intern at Baidu.

[2] Lin Li, Yufei Liang and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, P. R. China.

[*] Corresponding authors, email: wanguowei@baidu.com and yongliu@iipc.zju.edu.cn.

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160492

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

To summarize, our main contributions are:

- A joint learning framework for overlap estimation and 3D local features, which effectively fulfills loop closure and 6-DoF registration in urban scenes.
- A novel overlap estimation method that fully interacts with pairwise information, yielding high precision and recall under low overlap scenes.
- Based on BEV, the separation of 2D keypoint detection on the BEV plane and height regression makes it an efficient and practical 3D keypoint detection method.
- Rigorous tests and detailed ablations on the KITTI [16] and Apollo-Southbay [6] datasets to comprehensively verify the effectiveness of the proposed method.

II. Related Work

A. LiDAR-based Loop Closure Detection

As a hot research field of SLAM, numerous researchers have studied LiDAR-based loop closure detection, and many excellent works have been proposed [7], [10], [17]–[25].

One of the common solutions is encoding the input point cloud into a global descriptor [26], [27] as a 1D vector or 2D matrix and comparing their similarity to find the loop closures. Scan Context [9] globally describes the point cloud as a bird's-eye view (BEV) with height information in polar coordinates, which makes the descriptor robust to rotation and has good generalization. Extension works [10], [17]–[20] encode additional semantic [10] and intensity [17] information further to improve the detection performance.

Recently learning-based methods have shown impressive results. Some works [7], [21]–[25] extract local features with a deep network and aggregate them to a global descriptor with NetVLAD [28] or other context gating techniques [29]–[31]. Another common practice is to segment the point cloud into objects as local features and then match them directly [32], [33] or by a graph [34], [35].

The above methods only encode features from its single input point cloud and do not know about the correlated information from the counterpart one. Predator [12] fuses the feature maps from two stream networks and implicitly encodes the overlapping context with designated supervision loss to handle the low overlap registration problem. Unlike Predator, overlap estimation is an explicit network design in our method, rather than supervision only, which is more conducive to obtaining desired results. Furthermore, we use the deepest feature maps that contain contextual information for interacting instead of at the point level, which makes our method more robust and is validated in our experiments.

B. Deep Point Cloud Registration

Point cloud registration is also a widely studied topic in SLAM research society, in which deep learning methods demonstrate promising results when solving challenge cases, e.g., bad initialization or low overlap. Some of these methods are based on directly inferring correspondences, in which key points are extracted and described on local patches [36]–[38], then the accurate transformation is obtained with robust pose estimation, e.g., RANSAC [39] or weighted Procrustes [40]. In order to learn local features more effectively, point convolution backbones [41], [42] are adopted to extract dense features in a single forward process [11], [12], [43].

Instead of directly finding correspondences, some other methods estimate the transformation in an end-to-end manner. Some of them [44]–[46] build soft correspondences by learning patch features and use a differentiable weighted SVD module to compute the transformation. Others [14], [47], [48] directly use the extracted features to regress the transformation.

These learning-based methods give competitive results, but the performance drops drastically in small overlap scenes. This problem already draws the attention of many researchers, as [12], [13], [15] tried in indoor scenes. Our method only detects keypoints in the estimated overlap, thus avoiding wrong matches between non-overlapping regions.

III. Methodology

In this section, we describe the architecture of the proposed unified BEV model for 3D local features and overlap estimation in detail, as shown in Fig. 2. Inspired by DiSCO [30], the two input point clouds are first converted to multi-layer BEV representations and then fed into a 2D U-Net backbone to extract multi-scale features. The deepest feature maps of the two input point clouds interact with each other via a cross-attention module to measure their relevance, after which a classification head is used to calculate two overlap score maps. Meanwhile, the multi-scale features are fed into a description head, a detection head, and a regression head to obtain feature descriptors and 3D keypoints, respectively.

A. Dense Feature Description

We first divide the input point clouds $\mathcal{P}$ and $\mathcal{Q}$ into $H \times W \times C$ grids, in which each voxel is set to 0 or 1 depending on its occupancy. By treating each pillar of the grid as a C-dimension channel, the point clouds $\mathcal{P}$ and $\mathcal{Q}$ are then converted into BEV representations, denoted as $B_P \in \mathbb{R}^{H\times W\times C}$ and $B_Q \in \mathbb{R}^{H\times W\times C}$.

2D UNet Backbone. Instead of using 3D (sparse) convolutions or point convolutions, we directly apply 2D sparse convolutions on BEV representations to extract deep features. Concretely, we use a 2D UNet-like structure with skip connections and residual blocks in the encoder and decoder. Considering the sparsity of the inputs, 2D sparse convolutions can be used to speed up. After performing 2D convolutions on BEV representations, we can obtain the following multi-scale feature maps:

$$E_t^1, \dots, E_t^s = \mathcal{E}(B_t),\ t \in \{\mathcal{P},\ \mathcal{Q}\} \tag{1}$$

$$F_t^{s-1}, \cdots, F_t^1 = \mathcal{F}(E_t),\ t \in \{\mathcal{P},\ \mathcal{Q}\}, \tag{2}$$

where $\mathcal{E}$ and $\mathcal{F}$ represent the encoder and decoder of the backbone, respectively.

Description Head. The description head extracts feature descriptors D_t from the feature map F_t^1, which consists of a 1×1 convolution and normalization layer as follows:

$$D_t = \mathrm{Norm}_{\mathrm{L}_2}(\mathrm{Conv}_{1\times1}(F_t^1)),\ t \in \{\mathcal{P},\ \mathcal{Q}\}, \tag{3}$$

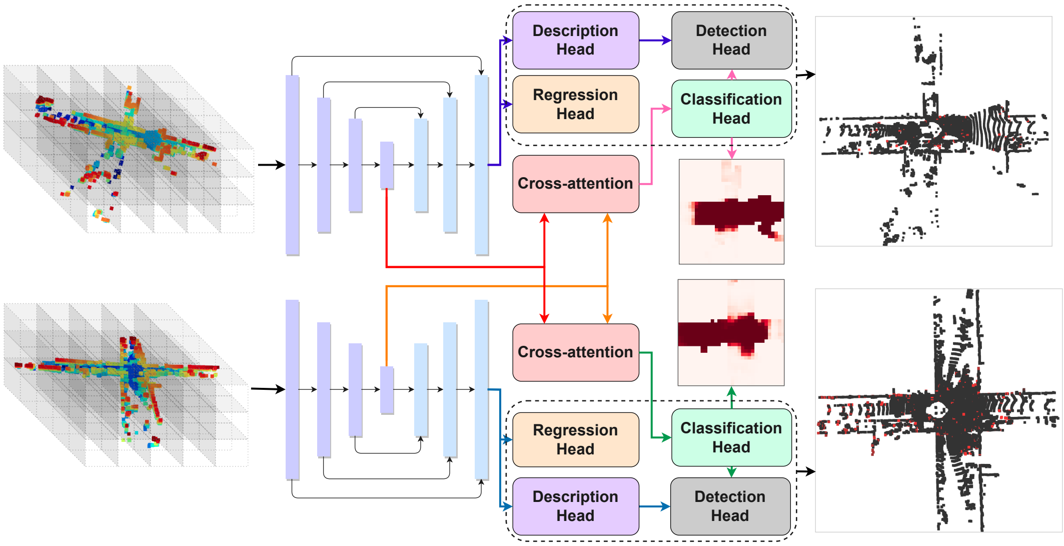

Fig. 2: The architecture of the proposed unified BEV model for 3D local features and overlap estimation. We take multi-layer BEV representations as the input of the 2D UNet backbone. The feature maps of the last layers of the encoder and decoder are used for overlapping region detection and local feature extraction, respectively.

where $\mathrm{Norm}_{\mathrm{L}_2}$ is the L_2 normalization operation across feature channels.

B. Dense Keypoint Detection

D3Feat [11] detects 3D keypoints of point cloud based on the loal maximum of the channel and spatial dimensions of the point features. Our keypoint detection also adopts a similar way, except that we detect 2D keypoints $\{(x,\, y)\}$ on the BEV feature descriptors D_t and regress their heights $\{z\}$ to form the 3D keypoints $\{(x,\, y,\, z)\}$, which makes our approach a more efficient implementation.

Detection Head. The spatial saliency of each pixel in D_t is evaluated in its local neighborhood. Thanks to the regularity of BEV representation, the neighborhood of each pixel simply consists of the pixels within the square centered around it, thus avoiding the heavy operation of kdtree search in D3Feat [11]. The spatial saliency score of each pixel p_{ij} is defined as

$$\alpha_{ij}^k = \ln\left(1 + \exp\left(D_{ij}^k - \frac{1}{|\mathcal{N}_{ij}|}\sum_{(i'j')\in\mathcal{N}_{ij}} D_{i'j'}^k\right)\right), \quad (4)$$

where $k = 1, ..., C$, and $\mathcal{N}_{ij}$ represents the non-empty neighboring pixels of p_{ij}.

For each pixel p_{ij}, there will be at most $s \times s$ non-empty pixels in its neighborhood, where s represents the length of the square. In this way, for Equ. 4, we can use the AvgPool operation to achieve an efficient implementation as follows.

$$\alpha_{ij}^k = \ln\left(1 + \exp\left(D_{ij}^k - \frac{s^2 \times \mathrm{AvgPool}(D)_{ij}^k}{s^2 \times \mathrm{AvgPool}(B^*)_{ij}}\right)\right), \quad (5)$$

where s is the window size of the average pooling, and $s^2 \times \mathrm{AvgPool}(D)_{ij}^k$ is the sum of k-th channel values of neighborhood, $B^* = \max_k(B^k)$ is the channel max value representing the occupancy of pillars, and $s^2 \times \mathrm{AvgPool}(B^*)_{ij}$ represents the number of the non-empty neighboring pixels. The s^2 in the numerator and denominator can be eliminated. In addition, a sparse AvgPool operation can be directly used to calculate the average of non-empty pixels in Equ. 4, thus replacing Equ. 5.

The channel max score is computed as

$$\beta_{ij}^k = \frac{D_{ij}^k}{\max_c D_{ij}^c}. \quad (6)$$

Both the spatial and channel scores are considered for computing the final detection score:

$$s_{ij} = \max_k(\alpha_{ij}^k \beta_{ij}^k). \quad (7)$$

Regression Head. The 2D salient keypoints can be detected in BEV representation with s_{ij}, but the heights still need to be recovered. Here we apply a regression head to predict a weight vector $W_{ij} \in [0,1]^{C\times 1}$ for each pillar B_{ij}. Let $H_{ij} \in \mathbb{R}^{C\times 1}$ denote the heights of voxels in a pillar, then the height of the keypoint in B_{ij} is predicted by a convolutional layer and a sigmoid layer as

$$W = \mathrm{Sigmoid}(\mathrm{Conv}_{3\times 3}(F_t^1)) \quad (8)$$

$$z_{ij} = W_{ij}^T * H_{ij}. \quad (9)$$

Finally, we obtain a regressed height z for each non-empty pillar and denote the regressed point clouds as $\mathcal{P}'$ and $\mathcal{Q}'$.

C. Overlapping Region Classification

Cross attention has demonstrated its effectiveness in interacting information [12], [13] and detecting overlap regions [15] from encoded feature maps. Similar to ImLoveNet [15], we adopt cross attention on two feature maps of the input point clouds to learn relevant information, followed by a classification head to solve the overlap as learning a similarity score. Different from [15], we only use the deepest feature maps for overlap estimation because the deepest feature maps contain richer context information and are easier to learn robust correlations.

Cross Attention. The cross attention module takes the deepest feature maps, $E^s_P \in \mathbb{R}^{H_s\times W_s\times C_s}$ and $E^s_Q \in \mathbb{R}^{H_s\times W_s\times C_s}$, to generate two relevant feature maps, $M_P \in \mathbb{R}^{H_s\times W_s\times C_s}$ and $M_Q \in \mathbb{R}^{H_s\times W_s\times C_s}$, in a bilateral way. Specific details are as follows.

$$\begin{aligned}M_P =& E^s_P + \mathrm{MLP}(\mathrm{cat}(E^s_P, \mathrm{att}(E^s_P, E^s_Q, E^s_Q)))\\ M_Q =& E^s_Q + \mathrm{MLP}(\mathrm{cat}(E^s_Q, \mathrm{att}(E^s_Q, E^s_P, E^s_P))),\end{aligned} \tag{10}$$

where $\mathrm{MLP}(\cdot)$ denotes a three-layer fully connected network, and att is the attention model, the detailed description can be referred from [15].

Classification Head. With the correlated feature maps M_P and M_Q from the cross attention module, we apply a binary classification to predict the overlap score maps, $\gamma_P \in [0,1]^{H_s\times W_s}$ and $\gamma_Q \in [0,1]^{H_s\times W_s}$, of $\mathcal{P}$ and $\mathcal{Q}$ as

$$\gamma_t = \mathrm{Sigmoid}\left(\mathrm{Conv}_{3\times 3}\left(\mathrm{ReLU}\left(\mathrm{Conv}_{3\times 3}\left(\mathrm{M_t}\right)\right)\right)\right), \tag{11}$$

where two 3×3 convolution layers, a sigmoid layer, and a ReLU are used.

Similarity Score. By counting the overlapped regions, we can obtain a similarity metric as

$$\tau = \frac{1}{2}\left(\frac{\sum \gamma_P}{V_P} + \frac{\sum \gamma_Q}{V_Q}\right), \tag{12}$$

where V_P and V_Q denote the number of occupied pixels of M_P and M_Q. In subsequent experiments IV-B, we will demonstrate that this similarity metric can be used for the loop closure detection task.

D. Loss Function

To train the network in an end-to-end manner, we utilize multi-task loss functions for jointly optimizing the feature description, keypoint detection, height regression, and overlap region classification.

Description Loss. Following [12], we take the circle loss [49] to learn discriminative descriptors. We perform random sampling to balance the number of positive and negative samples. The positive samples Ω_p are selected correspondences, where the set of correspondences is defined as points in $\mathcal{Q}'$ that lie within a radius around point i in $\mathcal{P}'$. The negative samples Ω_n are formed from points of $\mathcal{Q}'$ outside a larger radius of the point i. This loss function can be expressed by:

$$L^P_{desc} = \frac{1}{N}\sum_{i=1}^{N}\ln\left(1 + \sum_{j\in\Omega_p} e^{\theta^j_p(d^j_i-\Delta_p)} \cdot \sum_{k\in\Omega_n} e^{\theta^k_n(\Delta_n-d^k_i)}\right), \tag{13}$$

where Δ_p and Δ_n are positive and negative margins, d^j_i and d^k_i are feature distance of positive samples and negative samples, θ^j_p and θ^k_n are the positive and negative weights, computed for each sample individually with $\theta^j_p = \gamma(d^j_i - \Delta_p)$ and $\theta^k_n = \gamma(\Delta_n - d^k_i)$. We recommend referring to the original paper [49] for details. Through the same process, we can get the reverse loss L^Q_{desc}, and the total loss L_{desc} is the average of L^P_{desc} and L^Q_{desc}.

Detection Loss. The detection loss aims to encourage the easily matchable correspondences to have higher keypoint detection scores than the correspondences which are hard to match as

$$L_{det} = \frac{1}{N}\sum_i \left(d^{\mathrm{pos}}_i - d^{\mathrm{neg}}_i\right)\left(s_{P_i} + s_{Q_i}\right), \tag{14}$$

where (P_i, Q_i) are correspondences of $\mathcal{P}'$ and $\mathcal{Q}'$, s_{P_i} and s_{Q_i} denote their saliency scores, d^{pos}_i is the feature distance between positive samples, d^{neg}_i represents the feature distance between the hardest negative samples.

Regression Loss. For the input point cloud $\mathcal{P}$, to recover the heights of keypoints, we use both the origin point cloud $\mathcal{P}$ and its counterpart $\mathcal{Q}$ to supervise the recovered heights as

$$L^P_{reg} = \frac{1}{N}\sum_i \left(\|z_{P'_i} - z_{P_j}\| + \|z_{P'_i} - z_{Q'^T_i}\|\right), \tag{15}$$

where $z_{P'_i}$ is the predicted height of point P'_i in regressed point cloud $\mathcal{P}'$, z_{P_j} is the height of closest point P_j to P'_i in point cloud $\mathcal{P}$, and Q'^T is the point cloud $\mathcal{Q}'$ transformed into frame of $\mathcal{P}'$, $z_{Q'^T_i}$ is the predicted height of corresponding point of P'_i in the $\mathcal{Q}'^T$.

Classification Loss. A binary cross entropy is used in the classification loss as

$$L_{bce} = \mathrm{BCE}(\gamma_P, l_P) + \mathrm{BCE}(\gamma_Q, l_Q), \tag{16}$$

where BCE denotes the binary cross entropy, $l_P \in [0,1]^{H_s\times W_s}$ and $l_Q \in [0,1]^{H_s\times W_s}$ are ground-truth labels. In addition, we found that additional supervision of strengthening contrastive distance on the deepest feature map is helpful for network convergence. Therefore we construct another circle loss L_{sg} similar to Equ. 13 on the deepest feature map and forms the classification loss together with BCE loss L_{bce}.

IV. EXPERIMENTS

A. Datasets and Implementation Details

Our method has been extensively tested in real-world urban scenarios, primarily using two public datasets, the KITTI dataset [16], and the Apollo-SouthBay dataset [6], [39]. Code is available at https://github.com/lilin-hitcrt/BEVNet.

KITTI Odometry Dataset. The KITTI odometry dataset collected point clouds captured with a Velodyne HDL64 LiDAR. It contains a total of 22 sequences, of which only the first 11 have ground-truth pose annotations. We use the last 11 sequences to train our network and use the poses provided by semantic KITTI [50] for supervision. We validate the performance of our method in detecting loop closures on six sequences (00, 02, 05, 06, 07, 08). Like other methods, we verified the performance of point cloud registration on 08-10 sequences.

Apollo-SouthBay Dataset. The Apollo-SouthBay dataset collected point clouds using the same model of LiDAR as the KITTI odometry dataset, but in the San Francisco Bay area, United States. Similar to KITTI, it covers various scenarios, including residential areas, urban downtown areas, and highways. Our model is trained on sequence Columbia-Park

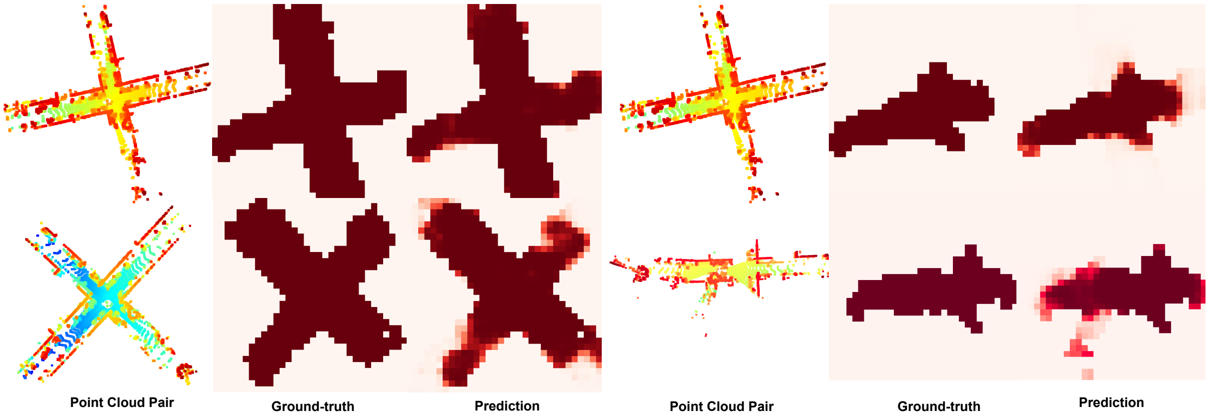

Fig. 3: We show the results of overlapping region detection on two pairs of point clouds with different distances. The first and fourth columns are the original point clouds, the second and fifth columns are the actual overlapping regions, and the third and sixth columns are the predicted overlapping regions.

TABLE I: Overlap classification results on KITTI dataset

Distance(m)	IOU(%)		Precision(%)		Recall(%)	
	OP [12]	Ours	OP [12]	Ours	OP [12]	Ours
10	73.5	**88.8**	91.5	**99.0**	78.9	**89.5**
20	65.9	**85.9**	82.4	**97.2**	76.6	**87.9**
30	56.3	**83.2**	70.0	**94.9**	74.3	**86.6**
40	42.5	**78.5**	52.8	**91.0**	69.6	**84.1**
50	29.5	**74.1**	35.5	**85.3**	67.1	**83.2**
60	15.9	**61.5**	18.1	**72.5**	63.5	**76.6**
Mean	47.3	**78.6**	58.4	**90.0**	71.7	**84.7**

The best scores are marked in bold.

TABLE II: Loop closure detection results on KITTI dataset

Methods	00	02	05	06	07	08	Mean
OT [25]	89.3	85.2	92.8	**100.0**	86.3	71.8	87.6
DS [30]	90.6	86.6	90.7	99.6	91.9	88.8	91.4
LC [52]	92.5	-	91.0	98.2	92.5	90.7	93.0
RI [29]	**97.5**	92.7	91.2	**100.0**	89.4	**97.5**	94.7
Ours	97.0	**94.5**	**97.2**	98.9	**95.7**	96.6	**96.7**

Average Recall@1. The best scores are marked in bold. [52] failed in sequence 02.

and tested on sequence Sunnyvale-Big-Loop to demonstrate the method's performance on small overlap loop closure detection and point cloud registration. Considering points are sparse in the region far from the LiDAR center in a single frame, we stitch point clouds from several consecutive frames into a *submap*. Additionally, the submap is cropped within 100m× 100m and voxelized into 50cm voxels for use.

Implementation Details. The BEV representation is formed into the shape of $256 \times 256 \times 32$. The backbone is configured with four layers in the encoder and three layers in the decoder, so the deepest feature, E^4, has the shape of $32 \times 32 \times 512$. In constructing the training data, point cloud pairs with distances ranging from 0 to 80 meters are used. We adopt spconv [51] to implement our backbone. Our code is based on PyTorch using the Adam optimizer with a learning rate of 10^{-4}.

B. Overlap Estimation

Overlapping Region Detection. As shown in Table. I, overlapping region evaluation metrics include intersection over union (IOU), classification precision and recall. On sequences 08-10 of the KITTI odometry dataset, different distances between pairs ranging from 10m to 60m are used to verify the influence of overlap size on our model's performance. The comparison method [12], marked as OP, was retrained with the same distance configuration. The result shows that our method can effectively detect overlapping regions with average IOU, precision, and recall of 78.6%, 90.0%, and 84.7%, respectively. As the distance between pairs is going large, the performance of Predator drops drastically, while our method shows better results in each distance range with only a slight fall, and still above 0.7 even under 60m distance. One of the causes is that overlap classification on the raw, unorganized point cloud in Predator is more difficult than in our method. The ablation studies IV-D have shown support for this view. Two pairs of point clouds and their ground truth and predicted overlapping regions are visualized in Fig. 3.

Loop Closure Detection. To make the model robust to loop closure detection, the issues of occlusion and small overlaps need to be addressed. To compare with the existing methods, Recall@1 in [30] is used as the evaluation metric. The best matching result for each query frame is inferred among the neighboring frames around the query, excluding 100 consecutive frames near the query. An inference is considered correct when its distance from the query is less than 10m. As shown in Table. II, our method outperforms the existing methods on most sequences. We can conclude that estimating the correct overlapping regions is the key factor in making our method stand out. Most methods could successfully detect loops in large overlapping scenes, while only our method survives with small overlaps, as illustrated in the right part of Fig. 3.

Loop closure detection is also tested on the Apollo-Southbay dataset. The tested pairs are sampled at various distances with 10m intervals. As shown in Tab. III, our method outperforms others at all distance settings. Since the Sunnyvale-Big-Loop sequence contains some quite dif-

TABLE III: Loop closure detection results on Apollo-SouthBay dataset

Methods	0-10m	10-20m	20-30m	30-40m	40-50m	50-60m
OT [25]	86.3	34.3	15.9	15.0	10.8	9.7
DS [30]	90.8	43.8	12.5	13.7	7.0	7.9
Ours	**97.9**	**85.7**	**62.5**	**66.0**	**57.3**	**57.1**

Average Recall@1. The best scores are marked in bold.

TABLE IV: Registration results on KITTI dataset

Methods	RTE(cm)	RRE(○)	RR(%)
3DFeat [53]	25.9	0.25	96.0
FCGF [43]	9.5	0.30	96.6
D3Feat [11]	7.2	0.30	99.8
SpinNet [54]	9.9	0.47	99.1
Predator [12]	6.8	0.27	99.8
COFiNet [55]	8.2	0.41	99.8
GeoTransformer [13]	6.8	0.24	99.8
Ours	7.5	0.26	99.8

ferent scenes which are not included in the training data, our method still works well, which demonstrates the better generalization capacity of our method.

C. Point Cloud Registration

For pairwise registration, state-of-the-art methods [11]–[13], [43], [53]–[55] are compared at a 10m distance setting, and all keypoints are used. We use RANSAC with 50,000 max iterations to estimate the transformation following [11]. As shown in Tab. IV, the relative translation error (RTE), relative rotation error (RRE), and registration recall (RR) [11] of our method are 7.5 cm, 2°, and 99.8%, respectively. The comparable accuracy our method achieved illustrates the effectiveness of our feature description and keypoint detection on BEV representations.

We conduct experiments with various distance settings to address the low overlap cases. With registration recall defined as RTE $<$ 2m and RRE $< 5°$, Predator [12] and our method with/without overlap estimation detection (O.w.O/O.w.o.O) are compared on the KITTI and Apollo-Southbay datasets. Up to 250 keypoints per point cloud are used. As shown in Tab. V, O.w.o.O achieves comparable results to OP [12] on KITTI, and O.w.O always keeps top performance, thus illustrating the usefulness of overlap estimation in small overlapping scenes. The comparison of our method shows the same conclusion on the Apollo-Southbay dataset, O.w.O/O.w.o.O methods give 77%/47% RR at 80m, respectively. The grouped comparison demonstrates that the overlap is crucial for registration: *A better overlap estimation makes better registration.*

TABLE V: Registration results at different distances

Distance(m)	KITTI			Apollo	
	OP [12]	O.w.o.O	O.w.O	O.w.o.O	O.w.O
10	97.1	**99.6**	**99.6**	99.5	**100.0**
20	95.4	96.8	**98.2**	99.8	**99.8**
30	80.5	87.0	**96.2**	99.3	**99.7**
40	51.1	59.9	**86.9**	97.7	**98.8**
50	24.5	33.0	**67.9**	93.0	**97.2**
60	9.4	11.8	**47.1**	88.2	**94.5**
70	-	-	-	74.2	**89.7**
80	-	-	-	47.4	**76.7**

Registration Recall (%). The best scores are marked in bold.

TABLE VI: Ablation study on registration

Loss	Registration Metrics		
	RR	RTE	RRE
L_{desc}	1.46	119.4	3.97
$L_{desc} + L_{reg}$	40.1	94.1	2.33
$L_{desc} + L_{reg} + L_{det}$	59.9	63.2	1.77
$L_{desc} + L_{reg} + L_{det} + L_{bce} + L_{sg}$	**86.9**	**57.0**	**1.63**
Loss	Overlap Metrics		
	OI	OP	OR
$L_{desc} + L_{reg} + L_{det} + L_{bce}$	60.2	74.5	75.0
$L_{desc} + L_{reg} + L_{det} + L_{bce} + L_{sg}$	**78.6**	**90.0**	**84.7**

OP: overlap estimation precision. OR: overlap estimation recall.
OI: overlap estimation IOU.
The best scores are marked in bold.

TABLE VII: Ablation study on overlap estimation

Feature map	Size	IOU (%)	Precision (%)	Recall (%)
F^1	256×256	50.3	62.2	74.2
F^2	128×128	64.9	78.2	78.7
F^3	64×64	73.8	85.2	83.2
E^4	32×32	**78.6**	**90.0**	**84.7**

The best scores are marked in bold.

D. Ablation Study

We ablate the loss functions for registration and overlap estimation tasks on the KITTI dataset. The ablation of registration uses a 40m distance setting for the point cloud pair and 250 keypoints for each point cloud, while the ablation of overlap estimation is the same as in Section IV-B. As shown in Tab. VI, height information supervised by L_{reg} is indispensable for registration recall. The loss $L_{bce}+L_{sg}$ for overlap estimation improves the registration accuracy significantly (recall+27%). The detection loss L_{det} helping to select discriminate key points also boosts recall with +20%, RTE with -30cm. In addition, the loss term L_{sg} has a remarkable influence on overlap estimation (precision+15%). Another ablation is performed on the choices of feature maps for overlap estimation. As shown in Tab. VII, the performance, including IOU, precision, and recall consistently increases as the size of feature maps becomes smaller. As mentioned in Section III-C, the deepest feature maps are the best scale to make overlap region classification.

V. CONCLUSION

We have presented a unified BEV model that jointly learns 3D local features and overlap estimation for point cloud registration and loop closure. The BEV representation makes it convenient to use a shared backbone for related multi-task processes. Overlap estimation plays a core role in significantly enhancing performance on both registration and loop closure, especially in low overlap scenarios. As a further extension of this work, we plan to add a new task head to generate a global descriptor that makes the method capable of place recognition at global scope retrieval. Furthermore, we will explore an end-to-end registration process that directly generates the relative transformation without RANSAC post-processing.

REFERENCES

[1] J.-E. Deschaud, "Imls-slam: Scan-to-model matching based on 3d data," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2480–2485, 2018.

[2] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014* (D. Fox, L. E. Kavraki, and H. Kurniawati, eds.), 2014.

[3] T. Shiratori, J. Berclaz, M. Harville, C. Shah, T. Li, Y. Matsushita, and S. Shiller, "Efficient large-scale point cloud registration using loop closures," in *2015 International Conference on 3D Vision*, pp. 232–240, 2015.

[4] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A robust pose graph approach for city scale lidar mapping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1175–1182, 2018.

[5] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, "Lidar scan feature for localization with highly precise 3-d map," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1345–1350, 2014.

[6] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: Towards learning based LiDAR localization for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6389–6398, 2019.

[7] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4470–4479, 2018.

[8] 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," *Autonomous Robots*, vol. 46, no. 1, pp. 61–81, 2022.

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

[10] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "Ssc: Semantic scan context for large-scale place recognition," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2092–2099, IEEE, 2021.

[11] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6359–6367, 2020.

[12] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4267–4276, 2021.

[13] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11143–11152, 2022.

[14] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "Omnet: Learning overlapping mask for partial-to-partial point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3132–3141, 2021.

[15] H. Chen, Z. Wei, Y. Xu, M. Wei, and J. Wang, "Imlovenet: Misaligned image-supported registration network for low-overlap point cloud pairs," in *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–9, 2022.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, 2012.

[17] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2095–2101, IEEE, 2020.

[18] Y. Fan, Y. He, and U.-X. Tan, "Seed: A segmentation-based egocentric 3d point cloud descriptor for loop closure detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5158–5163, IEEE, 2020.

[19] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Lidar iris for loop-closure detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5769–5775, IEEE, 2020.

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

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

[22] M. Y. Chang, S. Yeon, S. Ryu, and D. Lee, "Spoxelnet: Spherical voxel-based deep place recognition for 3d point clouds of crowded indoor spaces," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8564–8570, IEEE, 2020.

[23] Z. Zhou, C. Zhao, D. Adolfsson, S. Su, Y. Gao, T. Duckett, and L. Sun, "Ndt-transformer: Large-scale 3d point cloud localisation using the normal distribution transform representation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5654–5660, IEEE, 2021.

[24] P. Yin, L. Xu, J. Zhang, and H. Choset, "Fusionvlad: A multi-view deep fusion networks for viewpoint-free 3d place recognition," *Ieee Robotics and Automation Letters*, vol. 6, no. 2, pp. 2304–2310, 2021.

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

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

[27] K. P. Cop, P. V. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using lidar intensities," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3653–3660, IEEE, 2018.

[28] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

[29] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "Rinet: Efficient 3d lidar-based place recognition using rotation invariant neural network," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4321–4328, 2022.

[30] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, "Disco: Differentiable scan context with orientation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2791–2798, 2021.

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

[32] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272, IEEE, 2017.

[33] R. Dube, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[34] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3d point clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8216–8223, IEEE, 2020.

[35] Y. Zhu, Y. Ma, L. Chen, C. Liu, M. Ye, and L. Li, "Gosmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5151–5157, IEEE, 2020.

[36] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 602–618, 2018.

[37] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5545–5554, 2019.

[38] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1802–1811, 2017.

[39] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[40] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2514–2523, 2020.

[41] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3075–3084, 2019.

[42] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

[43] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8958–8966, 2019.

[44] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8893–8902, 2021.

[45] Y. Wang and J. M. Solomon, "Prnet: Self-supervised learning for partial-to-partial registration," *Advances in neural information processing systems*, vol. 32, 2019.

[46] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11824–11833, 2020.

[47] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7163–7172, 2019.

[48] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11366–11374, 2020.

[49] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, 2020.

[50] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[51] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[52] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, and C. Fookes, "Locus: Lidar-based place recognition using spatiotemporal higher-order pooling," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5075–5081, IEEE, 2021.

[53] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 607–623, 2018.

[54] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11753–11762, 2021.

[55] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23872–23884, 2021.