

SGT-LLC: LiDAR Loop Closing Based on Semantic Graph With Triangular Spatial Topology

Shaocong Wang^{ID}, Fengkui Cao^{ID}, Member, IEEE, Ting Wang^{ID}, Xieyanli Chen^{ID}, Member, IEEE,
and Shiliang Shao^{ID}, Member, IEEE

Abstract—Inspired by how humans perceive, remember, and understand the world, semantic graphs have become an efficient solution for place representation and location. However, many current graph-based LiDAR loop closing methods focus on extracting adjacency matrices or semantic histograms to describe the scene, which ignore a lot of multifaceted topology information for efficiency. In this letter, we propose a LiDAR loop closing method based on semantic graph with triangular spatial topology (SGT-LLC), which fully considers both semantic and spatial topological information. To ensure that descriptors contain robust spatial information while maintaining good rotation invariance, a local descriptor based on semantic topological encoding and triangular spatial topology is proposed, which can effectively correlate scenes and estimate 6-DoF poses. In addition, we aggregate local descriptors from various nodes in the graph using fuzzy classification to create lightweight database and efficient global search. Extensive experiments on KITTI, KITTI360, Apollo, MuRAN and MCD datasets prove the superiority of our approach, compared with state-of-art methods.

Index Terms—SLAM, localization, mapping.

I. INTRODUCTION

L OOP closure detection is a crucial component of SLAM systems, which can correct drift errors while ensuring a globally consistent map. Considering the illumination and scale invariance of LiDAR sensors, more and more researchers are

Received 29 September 2024; accepted 10 February 2025. Date of publication 17 February 2025; date of current version 26 February 2025. This article was recommended for publication by Associate Editor M. Hanheide and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62203091, in part by the China Postdoctoral Science Foundation under Grant GZB20230804, in part by the National Natural Science Foundation of China under Grant 62403458, in part by the Liaoning Natural Science Foundation under Grant 2024-B-SBA-49, and in part by the Autonomous Project of State Key Laboratory of Robotics under Grant 2024-Z09. (Corresponding authors: Fengkui Cao; Ting Wang.)

Shaocong Wang is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, also with the Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China, and also with the University of Chinese Academy of Sciences, Beijing 110003, China.

Fengkui Cao, Ting Wang, and Shiliang Shao are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: caofengkui@sia.cn; wangting@sia.cn).

Xieyanli Chen is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410003, China.

The code will be available at: <https://github.com/ROBOT-WSC/SGT-LLC.git>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3542695>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3542695

focusing on LiDAR-based loop closure detection and place recognition.

Traditional loop closure detection methods typically operate directly on the raw point clouds to extract features or perform registration [1], [2], [3]. While these methods can accurately detect scene correlations, they often require significant time or memory. To enhance efficiency, some researchers use vision-based approaches to create scene descriptors from projected images of point clouds [4], [5], [6], [7]. However, due to the dimensionality reduction inherent in these methods, they can only estimate 1-DoF or 3-DoF transformations between point clouds.

Inspired by the way humans remember scenes through their layouts, semantic topological graphs have emerged as an efficient method for scene representation. Current mainstream methods only use low-level topological relationships, such as node adjacency, while ignore more complex semantic spatial relationships [8], [9], [10]. This can lead to incorrect associations between semantic topological graphs in complex environments, resulting in false positive loop closure detection. Furthermore, these methods usually rely on complex geometric validation and optimization to conduct 6-DOF pose estimation after graph matching, which can significantly increase computational time and memory usage. Thereby, achieving lightweight, stable, and high-quality semantic topology matching remains a challenging problem.

To address these challenges, we propose an accurate, efficient loop closure detection method SGT-LLC, leveraging multifaceted information of the semantic graphs. The proposed method start by constructing a second-order semantic topology graph using only stable semantic labels present in the scene, while excluding unstable labels such as ground, vehicles, and pedestrians. Next, we introduce an effective local descriptor based on semantic topological hash encoding and triangular spatial topology, which ably combines semantic information with spatial information. These local descriptors are then processed using fuzzy classification to form a representative and lightweight global descriptor, enabling rapid global search. Moreover, our proposed descriptors effectively correlate 3D topological nodes across different scenes, enabling quick and precise 6-DoF pose estimation through object-level SVD.

The main contributions of our method are as follows:

- A novel descriptor based on semantic topological hash encode is proposed, which includes the triangular topological relationships extended from the second-order semantic topological nodes. Benefiting from the stable triangular structure, our method can accurately associate nodes between semantic graphs and estimate 6-DoF poses between

point clouds using object-level SVD without requiring any additional optimization.

- A global topological descriptor is created by aggregating local descriptors from various nodes through fuzzy classification, which supports lightweight place representation and efficient global search.
- The proposed loop closure detection system has undergone rigorous evaluation across multiple benchmark datasets, including KITTI, KITTI360, Apollo, MulRAN, and MCD. These datasets encompass both mechanical and solid-state LiDAR modalities. The experiments demonstrate that our system achieves better performance compared to the other state-of-the-art methods.

II. RELATED WORKS

A. Geometry-Based Methods

Early LiDAR-based loop closure detection methods primarily focused on directly matching similar scenes using raw point clouds registration. While effective, these methods often face efficiency limitations. To enhance algorithmic efficiency, researchers have increasingly focused on processing point clouds after projection. Scan Context [4] and its variants [5], [6] projects point clouds into Bird's Eye View (BEV) polar grids based on height or other information, which not only improves computational efficiency but also provides good rotation invariance. SOLID [7] projects point clouds onto range and angle images to address the limitations imposed by restricted field-of-view (FOV) scenarios. However, projection-based methods often estimate only 1-DoF (yaw) or 3-DoF (x, y, yaw) transformations between point clouds, failing to address the full 6-DoF loop closure pose. To overcome this limitation, some researchers have focused on constructing descriptors based on the geometric relationships of points in point clouds. Bow3D [1] combines a bag-of-words model with geometric features to enable the estimation of 6-DoF loop closure poses. STD [2] characterizes scenes using triangular structures in point clouds, achieving a balance between accuracy and real-time performance.

B. Topology-Based Methods

In recent years, inspired by how humans remember scenes through the distribution of object positions, semantic topological graph have become an efficient place representation method. Liao et al. [11] have incorporated additional semantic topological information into Scan Context for outdoor dynamic scenes, while this approach shares the same limitation as the Scan Context family: it cannot fully estimate 6-DoF loop closure poses. Outram [3] incorporates semantic assistance into STD to develop a precise global localization methodology; however, this approach exhibits high time complexity, thereby posing significant challenges for real-time execution. Currently, mainstream topology-based methods primarily focus on constructing semantic histograms or adjacency matrices to aid in matching topology graphs, and then determine node correspondences to estimate 6-DoF closed-loop poses [8], [9], [10]. Using semantic histograms to represent semantic topology is efficient, but these methods only considers node adjacency and neglects higher-level topological relationships. This limitation can lead to incorrect scene correlations in complex or large-scale scenarios, which is critical

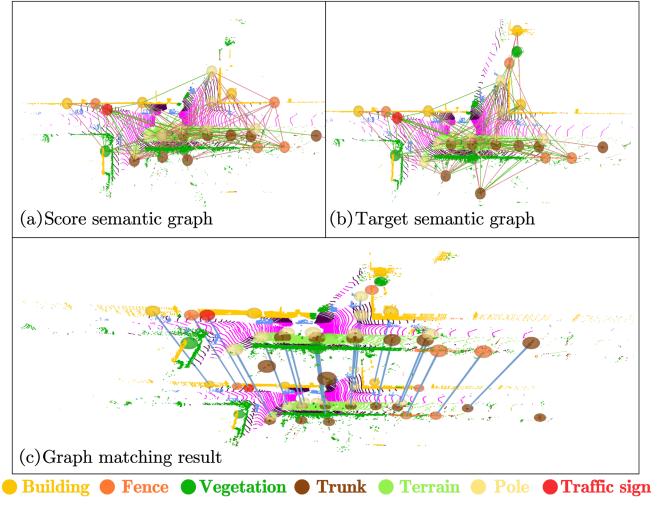


Fig. 1. The matching results of the similar scenes obtained by the proposed method. (a) The semantic graph of source point cloud. (b) The semantic graph of target point cloud. (c) The matching result of semantic graphs, the blue line connects the associated nodes.

for loop closure detection systems. Consequently, these methods necessitate complex geometric validation and optimization to mitigate the effects of unreliable node matching, which can substantially increase the time and space complexity of the algorithm.

Unlike the above methods, this letter proposes a novel loop closure detection method based on semantic graph with triangular spatial topology. It is noteworthy that the proposed method achieves simultaneous place recognition and pose estimation through a concise and robust descriptor matching mechanism without requiring any additional optimization, thereby significantly reducing both time and spatial complexity. Our method leverages second-order semantic graph and encodes descriptor using distinctive semantic hashing and triangular topological relations. This results in a lightweight, accurate, and robust place representation.

III. THE PROPOSED METHOD

The framework of SGT-LLC is illustrated in Fig. 2, comprising three key components: Semantic Topology Graph Construction, Descriptor Extraction, Graph Matching and Pose Estimation.

A. Semantic Topological Graph Construction

1) *Semantic Segmentation*: Our method is highly compatible with mainstream point cloud semantic segmentation frameworks such as RangeNet++ [12], PolarNet [13], and Spvans [14]. To improve the network's computational efficiency, we use TensorRT for acceleration, ensuring seamless integration with the existing C++-based SLAM systems. For semantic labels, we filter less representative categories such as ground and unstable categories like vehicles and pedestrians. Finally, seven stable categories (Building, Fence, Vegetation, Trunk, Terrain, Pole, Traffic sign) as shown in Fig. 1 are selected to establish the semantic topology graphs.

2) *Node Extraction*: In methods based on semantic topology graphs, the Euclidean Cluster algorithm is commonly employed

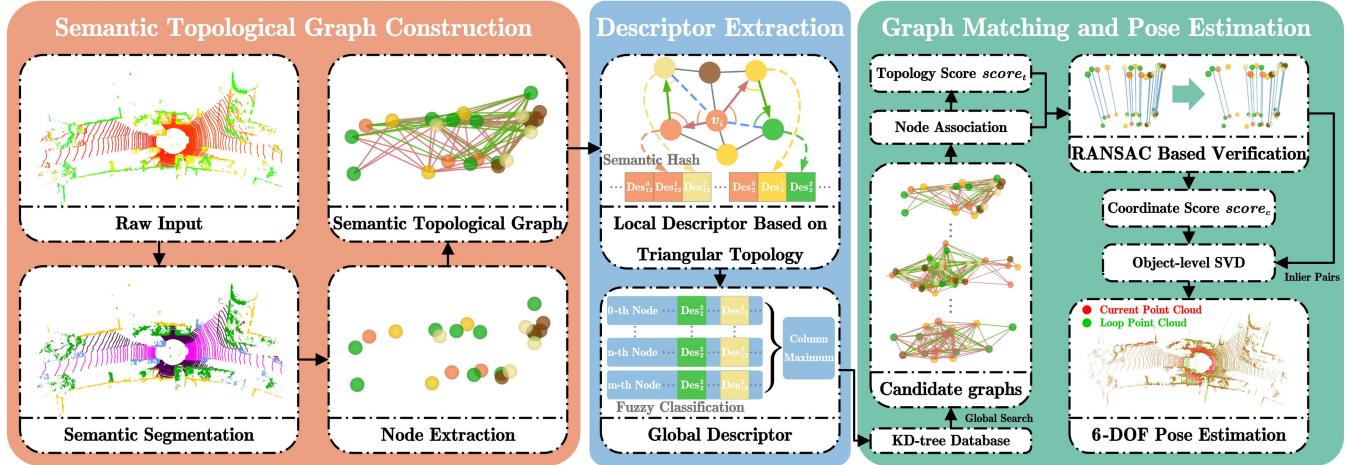


Fig. 2. System overview of SGT-LLC. First, we extract the second-order semantic topology for representative labels from the segmented point cloud using clustering. Next, we encode the descriptors using semantic topology and incorporate stable spatial information through triangular spatial topology. To create lightweight database and efficient global search, we aggregate the values of local descriptors from all nodes in the scene. Then we associate the candidate scenes obtained by global search with the current scene through a special node similarity comparison. Finally, RANSAC is used to refine the matched node pairs, and the similarity score between scenes is calculated. If the score is higher than threshold, the 6-DoF pose can be estimated based on the inlier nod pairs and object-level SVD.

to extract nodes, just like [8], [9], [10]. However, traditional Euclidean Cluster algorithm is time-consuming, which brings challenges for real-time loop closure detection systems. In this section, we adopt the Fast Euclidean Clustering (FEC) [15] to extract semantic nodes, significantly reducing node extraction time. Using FEC clustering on points of the same category after semantic segmentation, we obtain the node set $V = \{v_i\}$. Each node v_i includes both the cluster center $c_i = \{x_i, y_i, z_i\}$ and the semantic label l_i . Note that, to ensure a balanced number of nodes with different labels, we assign varying clustering radius and minimum clustering number to point cloud sets with different labels.

3) Second-Order Semantic Topology Construction: To improve the specificity of different semantic topology graph, we don't construct the graph directly from the adjacency matrix, as done in [8], [9], [10], but instead establish a second-order semantic topology graph to represent related scenes. For node v_i , a topological matrix T_i can be used to store the second-order topological relations originating from v_i . Compared to traditional first-order semantic graph, second-order semantic graph increases the diversity of edges while enhancing the distinction between nodes with the same label.

Firstly, we construct a adjacency matrix A for nodes in the scene based on the Euclidean distance between node pairs. If the Euclidean distance of node pair (v_p, v_q) is less than distance threshold σ_d , $A[p][q]$ is set to 1; otherwise, it is set to 0. Next, we create a topological matrix T_i of size 7×7 based on the number of semantic labels for node v_i . $T_i[m][n]$ contains nodes v_j and v_k , where v_j is the first-order semantic topological node of v_i in the $l_i \rightarrow m \rightarrow n$ semantic configuration, and v_k is the second-order semantic topological node of v_i in the same configuration. Nodes v_j and v_k must satisfy the following conditions:

- v_i and v_j are first-order neighbors, while v_j and v_k are second-order neighbors. This implies that: $A[i][j] = 1$, $A[j][k] = 1$.
- The labels of v_j and v_k are m and n , respectively.

- In light of the lightweight and specificity of descriptors, we employ path length as a criterion to screen the topological structure. The path length of $v_i \rightarrow v_j \rightarrow v_k$ should be the longest among all second-order semantic topological paths starting from v_i with a semantic configuration of $l_i \rightarrow m \rightarrow n$.

where, the second-order topological path of $v_i \rightarrow v_j \rightarrow v_k$ is defined as $d_{ij} + d_{jk}$, d_{ij} is the distance between v_i and v_j , d_{jk} is the distance between v_j and v_k . If there is no second-order topological path with $l_i \rightarrow m \rightarrow n$ semantic configuration originates from v_i , then $T_i[m][n]$ is empty.

B. Descriptor Extraction

1) Local Descriptor Based on Semantic Triangular Topology: Following the construction of the topological matrix T_i , a local descriptor Des_i associated with node v_i can be generated. This descriptor effectively integrates spatial topological relationships with semantic information.

Firstly, for local descriptors, it is important that they have a consistent vector dimension. A consistent vector dimension facilitates more accurate similarity calculations when matching descriptors corresponding to different scenes. To improve the accuracy of similarity calculations between descriptors, each dimension of the descriptors corresponding to different nodes should convey the same meaning. Therefore, inspired by the approach in [9], we use a semantic hash encoding to define the dimensions of descriptor and the meanings of each dimension. Given that the semantic label of v_i is determined, we use all possible second-order semantic topological configurations of v_i as the basis for hash encoding, constructing a 49-dimensional (7×7) semantic topological vector Des_i , where the numeral "7" denotes the number of semantic labels employed in our method. The index α corresponding to the semantic configuration $l_i \rightarrow m \rightarrow n$ in Des_i is defined as follows:

$$\alpha = 7 \times m + n \quad (1)$$

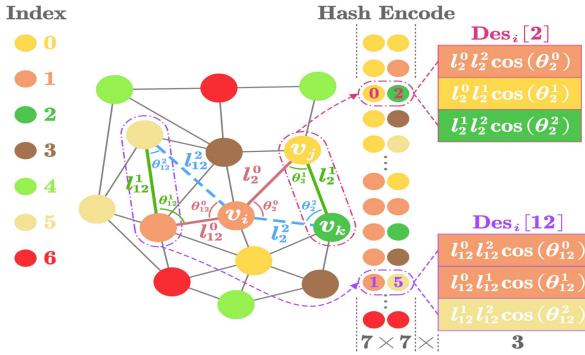


Fig. 3. An example for creating local descriptors. Des_i is encoded by various second-order semantic configurations. To enhance the stability and specificity of the spatial topological information in $Des_i[\alpha]$, we extend the second-order topological relations to triangular spatial topology for encoding $Des_i[\alpha]$.

To enhance the stability and specificity of the spatial topological information contained in $Des_i[\alpha]$, we did not merely use distance or angle information for encoding. Instead, we developed a triangular topological sub-descriptor to encode the geometric relationships of v_i under different topological configurations. $Des_i[\alpha]$ includes a 3-dimensional sub-vector that encodes the spatial topological relationship between v_i , v_j and v_k , where v_j and v_k are semantic topological nodes stored in $T_i[m][n]$. While constructing an edge between v_i and v_k , these three nodes naturally form a triangular topology with corresponding side lengths $\{l_\alpha^0, l_\alpha^1, l_\alpha^2\}$ and angles $\{\theta_\alpha^0, \theta_\alpha^1, \theta_\alpha^2\}$. To incorporate both angle and distance information into the descriptors, we employed a vector dot product method to construct $Des_i[\alpha]$. The different dimensions in $Des_i[\alpha]$ represent the dot product of adjacent vectors when v_i , v_j and v_k serve as the root node, respectively. If $T_i[m][n]$ is empty, $Des_i[\alpha]$ will be set to a zero vector. Fig. 3 illustrates the construction process for $Des_i[2]$ and $Des_i[12]$, they correspond to two semantic configurations: $1 \rightarrow 0 \rightarrow 2$ (*Fence* \rightarrow *Building* \rightarrow *Vegetation*) and $1 \rightarrow 1 \rightarrow 5$ (*Fence* \rightarrow *Fence* \rightarrow *Pole*).

2) *Global Descriptor for Retrieval*: We aggregate the values of local descriptors from all nodes in the scene to derive global descriptor for efficient global search and place representation. The global descriptor Des^G encodes all possible second-order semantic topological configurations in the scene using semantic hash encoding. To minimize memory usage and reduce database search time, the labels of first-order topological nodes are simplified into *foreground* and *background* using fuzzy classification rule. Under this rule, smaller objects such as trunks, terrain features, poles, and traffic signs are classified as *foreground*, whereas larger objects like buildings, fences, and vegetation are classified as *background*. After the simplification, the dimension of Des^G is reduced from $343(7 \times 7 \times 7)$ to $98(7 \times 2 \times 7)$. Based on the concept of max pooling, $Des^G[\beta]$ is defined as follows:

$$\begin{cases} \beta = 14 \times o + 7 \times p + q \\ Des^G[\beta] = \max_{l_i=o, m \in F[p], n=q} (Des_i[\alpha][2]) \end{cases} \quad (2)$$

where, β is the index of semantic configuration $o \rightarrow p \rightarrow q$, $p \in \{0, 1\}$; α is defined in (1); $F = [\text{foreground}, \text{background}]$ is the fuzzy collection of semantic labels.

Algorithm 1: Topological Node Association.

```

Input:  ${}^cV$ ,  ${}^hV$ : node sets of two graphs;
        ${}^cDes$ ,  ${}^hDes$ : descriptor sets of two graphs;
Output:  $LA$ : list of associated node pairs;
1 Let  $S$  be a zero matrix of size  $M \times N$ ;
2 foreach  ${}^cDes_i$ ,  ${}^hDes_j$  in  ${}^cDes$ ,  ${}^hDes$  do
3   if  ${}^cL_i = {}^hL_j$  then
4     for  $k = 0$  to  $48$  do
5       if  $\cos({}^cDes_i[k], {}^hDes_j[k]) > \sigma_1$  then
6         |  $S[i][j] = S[i][j] + 1$ 
7       end
8     end
9   end
10 end
11 for  $i = 0$  to  $M - 1$  do
12   Let  $max\_score = 0$  ;
13   for  $j = 0$  to  $N - 1$  do
14     if  $S[i][j] > max\_score$  then
15       |  $max\_score = S[i][j]$ 
16     end
17   end
18   if  $max\_score > \sigma_2$  then
19     | Add  $({}^cV_i, {}^hV_j)$  into  $LA$ 
20   end
21 end

```

C. Graph Matching and Pose Estimation

In this process, we first search the 15 historical scenes most similar to the Des^G in current scene from the kd-tree as candidate graphs. Then, we conduct a fine-grained analysis of these scenarios.

1) *Node Association*: Assume that the current graph contains M semantic nodes, with cV and cDes representing the node and descriptor sets, respectively. A candidate graph contains N semantic nodes, with hV and hDes representing the node and descriptor sets, respectively. Subsequently, the list of associated node pairs LA in cV and hV can be obtained by Algorithm 1. Firstly, we traverse all matching combinations of nodes in cV and hV , calculating the similarity score for each node pair with the same semantic label by comparing their descriptor differences. Note that the descriptor we constructed in III-B1 is a 147-dimensional sparse vector. Due to the sparsity of high-dimensional vectors, directly calculating the Euclidean or cosine distance between two descriptors can not effectively reflect their similarity. Therefore, instead of operating on the descriptor as a whole, we construct the similarity score by counting the number of sub-vector pairs in the two descriptors whose cosine similarity exceeds the threshold σ_1 . The similarity score between semantic nodes is represented as a similarity matrix S with size $M \times N$. We then traverse each row of S , and if the maximum score in a row exceeds the threshold σ_2 , the corresponding node pair is added to LA .

After extracting the LA corresponding to the current scene and all candidate scenes, we first calculate the node topology similarity score $score_c$ for initial screening of candidate scenes. $score_c$ is defined as the ratio of the LA 's length to the cV 's length. If $score_c$ exceeds a threshold σ_3 , we apply RANSAC to eliminate outliers in LA , with the proportion of inlier points serving as the node coordinate similarity score $score_t$.

TABLE I
F1 MAX[%]/EP[%] AND RTE [M]/ RRE [DEGREE] ON KITTI AND KITTI360

Methods	KITTI					KITTI360					
	00	02	05	08	Mean	00	04	05	09	Mean	
F1 MAX / EP	SC	85.7/69.6	64.3/61.6	94.5/85.8	32.8/58.1	69.3/68.8	76.5/75.6	71.4/56.9	81.5/68.2	81.2/66.9	77.6/66.9
	SSC	96.6/88.2	72.2/74.1	97.8/97.4	81.1/82.0	86.9/85.4	91.7/68.7	93.0/90.2	91.6/ 71.9	94.5/ 85.5	92.7/79.1
	CC	97.2/88.0	81.3/68.7	98.7/98.2	94.6/85.5	92.9/85.1	80.9/73.0	94.9/87.9	93.4/ 68.4	96.3/67.7	91.3/74.2
	SOLID	83.0/71.9	61.2/66.1	82.7/72.1	44.9/61.2	67.9/67.8	86.3/79.9	66.7/58.2	70.3/66.1	75.9/66.6	74.8/67.7
	STD	93.1/61.2	43.0/59.9	91.2/75.2	50.4/54.5	69.4/62.7	96.9 /83.6	70.1/50.6	68.2/60.2	87.8/65.9	80.8/65.1
	Outram	93.2/78.1	42.1/60.1	90.4/75.4	49.3/54.2	68.7/66.9	95.3/76.1	63.4/50.1	70.9/64.6	84.8/61.3	78.6/63.0
	BoW3D	96.3/85.8	36.9/53.9	95.5/70.2	39.0/50.9	66.9/65.2	94.6/76.5	80.1/53.6	80.5/56.1	90.4/68.6	86.4/63.7
	TGH	98.6/97.7	70.1/60.8	96.9/96.1	80.1/82.4	86.40/84.3	92.8/ 85.7	84.5/55.1	81.9/61.4	93.1/72.3	88.1/68.6
	SGLC	99.6/50.0	71.2/51.8	95.3/95.5	90.9/91.6	89.3/72.2	92.8/70.3	79.2/63.9	95.1/63.8	89.8/70.2	
Ours-R	99.7/ 99.5	83.0/75.5	99.7/99.4	95.4/94.3	94.5/92.2	95.5/84.3	97.1 / 91.9	94.2 /66.8	97.6 /79.4	96.1 / 80.6	
Ours-P	99.5/98.8	82.8/73.7	99.2/ 99.4	93.1/93.4	93.6/91.3	95.6 /84.1	96.7/90.6	92.5/64.2	96.8/75.5	95.4/78.6	
Ours-S	99.8 / 99.5	84.1 / 76.2	99.7 / 99.4	95.7 / 94.5	94.8 / 92.4	95.2/84.3	97.0/ 91.4	93.8/65.5	97.1/78.2	95.7/79.8	
Ours-G	99.8 /99.4	85.3 / 77.1	99.7 /99.3	95.8 / 94.7	95.1 / 92.6	95.5/ 84.5	97.4 /91.2	94.1 /66.3	98.2 / 81.2	96.3 / 80.8	
RTE / RRE	STD	0.78/1.28	2.12/1.32	0.28/1.05	1.86/2.39	1.26/1.51	0.28 /1.28	0.62/3.31	1.06/1.74	0.35/1.46	0.57 /1.94
	Outram	0.80/1.47	2.08/1.32	0.29/1.07	1.85/2.39	1.25/1.56	1.23/1.89	1.27/1.99	0.33 / 1.30	1.32/2.58	1.03/1.94
	BoW3D	0.75 /0.76	1.90 / 0.86	0.20 / 0.55	1.79 /2.52	1.16 / 1.17	0.78/ 1.18	1.55/ 1.65	0.98/1.44	0.11 / 0.92	0.85/ 1.29
	TGH	0.79/ 0.72	1.93/2.82	0.74/3.18	2.14/2.73	1.40/2.36	1.02/4.51	0.83/3.76	0.54/2.64	0.38/1.63	0.69/3.13
	SGLC	0.79/0.89	3.12/1.88	0.22/0.63	2.07/ 1.60	1.55/1.25	0.33/4.59	0.35 /2.25	1.13/3.83	1.46/2.35	0.82/3.25
Ours	0.71 / 0.32	1.81 / 1.32	0.13 / 0.09	0.92 / 0.82	0.89 / 0.63	0.17 / 0.64	0.19 / 0.67	0.25 / 1.00	0.15 / 0.42	0.19 / 0.68	

* Red and blue fonts denote the first and second place, respectively.

Fig. 5(a)–(b) shows the matched node pairs from different viewpoints, and (c)–(d) shows the filtering effect of RANSAC. This demonstrates that our method exhibits strong rotation invariance.

2) *6-DoF Pose Estimation*: We select the candidate scene with the highest $score_t$ as the loop closure scene when the highest $score_t$ exceeds the threshold σ_4 . Finally, we employ SVD on the inlier node pairs to obtain a 6-DoF pose estimation between the current scene and the loop closure scene. Note that, since the nodes are obtained from point cloud clustering, our pose estimation method operates at the object level.

IV. EXPERIMENTS

A. Experimental Setup

1) *Dataset*: For semantic segmentation, we use the SemanticKITTI dataset [16] to train the Networks, accelerating them with TensorRT to enable real-time application in loop closure detection systems.

The KITTI [17], KITTI360 [18], Apollo [19], and MuL-RAN [20] datasets are selected for evaluating the algorithm's performance with mechanical LiDAR(HDL-64, OS1-64). The MCD [21] dataset is selected for evaluating the algorithm's performance with solid-state LiDAR(Livox Mid-70). The solid-state LiDAR employed in MCD features a narrow 70° FOV, posing a significant challenge for conventional loop closure systems.

2) *Comparisons*: Nine state-of-the-art methods are selected for comparative experiments: Scan-Context(SC) [4], Semantic-Scan-Context(SSC) [5], Contour-Context(CC) [6], SOLID [7], STD [2], Outram [3], BoW3D [1], TGH [10] and SGLC [8]. Among these, STD, Outram, BoW3D, TGH and SGLC have the capability to estimate 6-DoF poses. SSC, Outram, TGH and SGLC are all need the assistance of semantic information.

B. Loop Closure Evaluation

In this section, we quantitatively analyze the place recognition and pose estimation capabilities of all methods. In the

experiment, loop closure pairs with a Euclidean distance of less than 3 m are considered positive samples, while pairs with a distance larger than 20 m are considered negative samples. For each method, we use its corresponding global search approach to identify the 15 most similar point clouds for each frame, calculating similarity to build a sample set for real-time evaluation.

1) *Place Recognition Performance*: When evaluating place recognition, we use the maximum F1 score ($F1 \max = \max(2 \times \frac{P \times R}{P+R})$) and Extended Precision ($EP = \frac{1}{2}(P_{R_{\min}} + R_{P_{100}})$) as evaluation metrics, where P and R means precision and recall. The statistical results are presented in the upper part of Tables I and II.

Mechanical LiDAR: Our method achieved superior performance across the majority of sequences in the KITTI, KITTI360, Apollo, and MuL-RAN datasets. The semantic information utilized in the baseline was derived from RangeNet++. To evaluate the impact of semantic labels from various sources on our algorithm's performance, we conducted experiments on the KITTI and KITTI360 datasets using labels predicted by RangeNet++ (Ours-R), PolarNet (Ours-P), Spvans (Ours-S), and Ground Truth (Ours-G). The results indicate that our method exhibits robust adaptability to different semantic segmentation networks.

Solid-state LiDAR: Due to the extremely sparse point cloud of Mid-70, we refer to STD, and the point cloud will be superimposed for loop closure detection. In the MCD dataset, most methods fail to achieve satisfactory results due to the narrow field of view of the solid-state LiDAR. However, our method, leveraging its robust node association mechanism, demonstrates superior performance even under such field-of-view limitations.

Fig. 4 shows the loop closure performance in two challenging intersection environments. It is evident that our method identifies a significantly higher number of loop closures compared to TGH and SGLC in scenarios with low overlap regions. This satisfactory performance is attributed to our descriptors, which incorporate richer semantic and spatial topological information, as opposed to relying solely on semantic histograms and adjacency matrices for place recognition.

2) *6-DoF Pose Estimation Performance*: We compared five 6-DoF pose estimation methods: STD, Outram, BoW3D, TGH,

TABLE II
F1 MAX[%]/EP[%] AND RTE [M]/ RRE [DEGREE] ON APOLLO, MULRAN AND MCD

Methods	Apollo 04			MulRAN				MCD Livox Mid-70				
	01	04	Mean	DCC-01	KAIST-01	Riversid-01	Mean	NTU-01	KTH-06	TUHH-02	Mean	
<i>F1 MAX / EP</i>	SC	99.2/99.2	96.5/94.1	97.8/96.6	87.6/85.1	99.2/98.7	68.6/67.9	85.1/83.9	52.8/60.3	5.8/50.0	60.5/56.7	39.7/55.6
	SSC	99.1/99.2	99.0/ 98.8	99.05/ 99	91.2/77.8	97.9/97.2	77.8/75.6	88.9/73.5	56.8/56.7	0.01/50.0	78.6/ 66.7	45.1/57.8
	CC	99.9/100	91.3/88.6	95.6/94.3	95.6/91.5	95.9/94.4	79.1/75.6	90.2/87.2	69.6/54.1	25.5/50.0	52.1/50.0	49.1/51.4
	SOLID	43.9/63.1	85.5/70.1	64.7/66.6	71.9/67.0	86.1/70.2	70.3/69.4	76.1/68.8	51.5/60.9	45.1/ 58.9	70.2/61.1	55.6/ 60.3
	STD	67.6/61.6	96.6/65.6	82.1/63.6	79.2/66.8	74.4/50.0	41.5/51.6	65.0/56.1	75.2/70.2	42.7/51.9	29.1/52.9	49.0/58.3
	Outram	86.9/87.9	93.5/56.1	90.2/72.0	79.3/66.8	80.9/54.8	57.6/57.6	72.6/59.7	74.5/68.7	34.2/51.4	54.9/57.3	54.5/59.1
	BoW3D	77.9/72.0	95.1/76.1	86.5/74.1	91.9/82.1	64.6/59.6	79.7/77.9	78.7/73.2	-	-	-	-
<i>RTE / RRE</i>	TGH	94.8/94.9	98.8/96.9	96.8/95.9	90.2/70.6	92.2/65.5	74.2/60.8	85.5/65.6	78.9/75.9	85.2 /50.0	80.5 /50.0	81.5 /58.6
	SGLC	99.9/100	99.1 /50.2	99.5 /75.1	89.8/64.8	92.5/93.1	85.9/82.2	89.4/80.0	78.7/55.8	3.0/50.6	24.8/50.3	35.5/52.2
Ours	99.9/100	99.7/99.7	99.8/99.85	93.3/93.7	99.2/99.2	91.7/92.3	94.7/95.1	90.4/91.2	98.2/98.2	82.5/73.8	90.4/87.7	
STD	0.23/1.11	0.47/1.62	0.35/1.37	0.33/1.31	0.30/ 1.23	0.36/1.36	0.33/1.30	1.04/2.42	1.42/4.26	0.58/3.08	1.01/ 3.25	
Outram	0.24/1.11	1.78/4.99	1.01/3.05	0.30/1.31	0.28/1.26	0.24/1.43	0.27/1.133	0.90/2.41	1.69/5.32	0.21/2.96		0.93/3.56
BoW3D	1.45/3.57	1.09/3.07	1.27/3.32	0.15/0.39	0.11/0.38	0.06/0.29	0.11/0.35	-	-	-	-	-
TGH	0.43/2.45	0.26/0.86	0.35/1.65	4.33/8.87	0.56/2.47	0.84/4.00	1.91/5.11	1.47/5.54	0.69/4.15	0.56/6.29	0.91/5.33	
SGLC	0.14/0.80	0.21/1.18	0.18/0.99	0.30/1.45	0.32/1.44	1.01/8.43	0.54/3.77	1.12/5.03	0.26/3.69	0.27/4.62	0.55/4.45	
Ours	0.05/0.04	0.36/2.18	0.20/1.11	0.21/0.46	0.27/1.68	0.20/1.13	0.23/1.09	0.49/1.92	0.16/2.34	0.28/ 1.31	0.31/1.85	

* Red and blue fonts denote the first and second place, respectively.

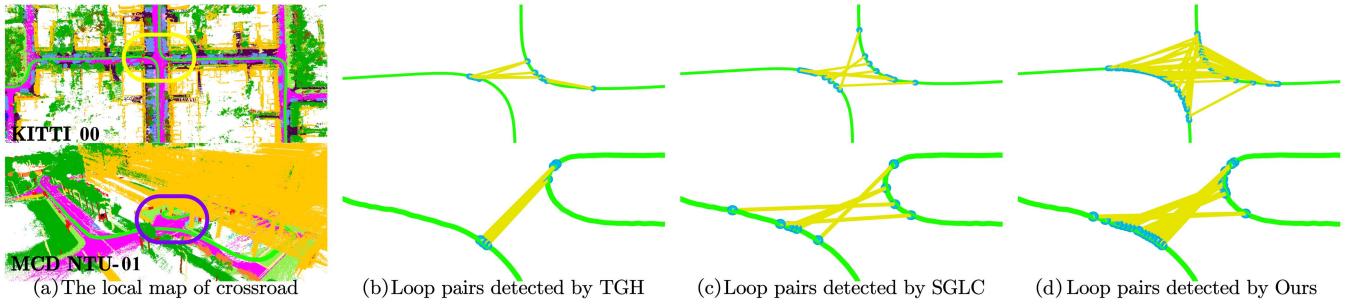


Fig. 4. Loop closure pairs detected by different methods in challenging intersection environments.

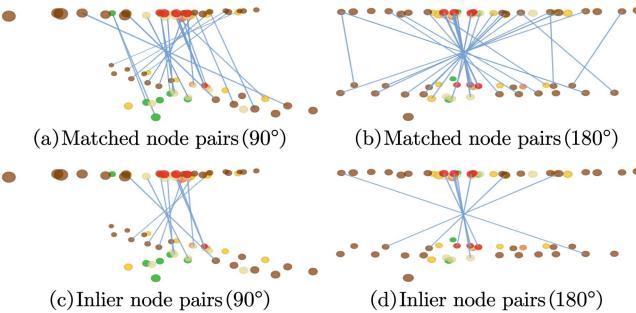


Fig. 5. The node matching result with different viewpoint transformations. (a)-(b) display the associated node pairs from III-C. (c)-(d) show the inlier node pairs after RNASAC.

and SGLC. Performance was evaluated using relative translation error (RTE) for translation accuracy and relative rotation error (RRE) for rotation accuracy. As shown in the lower half of Tables I and II, our method performed excellently on most sequences. Fig. 6 illustrates point cloud registration results for TGH, SGLC, and our method in four extreme cases. Panels (a) and (b) show mechanical LiDAR results under high rotation and low overlap, where both TGH and SGLC failed, but our method accurately estimated pose transformations using high-quality node associations. Panels (c) and (d) demonstrate solid-state LiDAR results with limited field of view, where our method had minimal ghosting after registration.

C. Loop Closure Distance Test

For loop closure detection system, the ability to reliably detect long-distance loop scenarios while maintaining overall system performance is a key measure of robustness. In this experiment, we evaluated this capability by varying the discrimination threshold for positive samples, and we selected four thresholds(3 m, 5 m, 10 m, 15 m) for test. To prevent mistakenly labeling distant scenes as positive samples, a pair is only considered positive sample if the Euclidean distance between scenes is below the threshold and the corresponding point cloud passes the ICP overlap rate check.

We tested two sequences: KITTI360 00 and MCD NTU-01, comparing our method with TGH and SGLC. As shown in Fig. 7, our method shows slower contraction of the precision-recall curve as the threshold distance increases, indicating greater robustness. The proposed descriptor integrates rich semantic and spatial topological information, thereby providing more stable and higher-quality graph correlations compared to TGH (which is limited to topological histograms) and SGLC (which considers only adjacency matrices).

D. Computation Efficiency Test

For a SLAM system, as it continues to operate, the database size grows continuously, leading to increased time and memory requirements for global searches. We conducted a statistical analysis of compute efficiency changes over time using the largest-scale sequence, KITTI360 09 (1447 s). For comparative

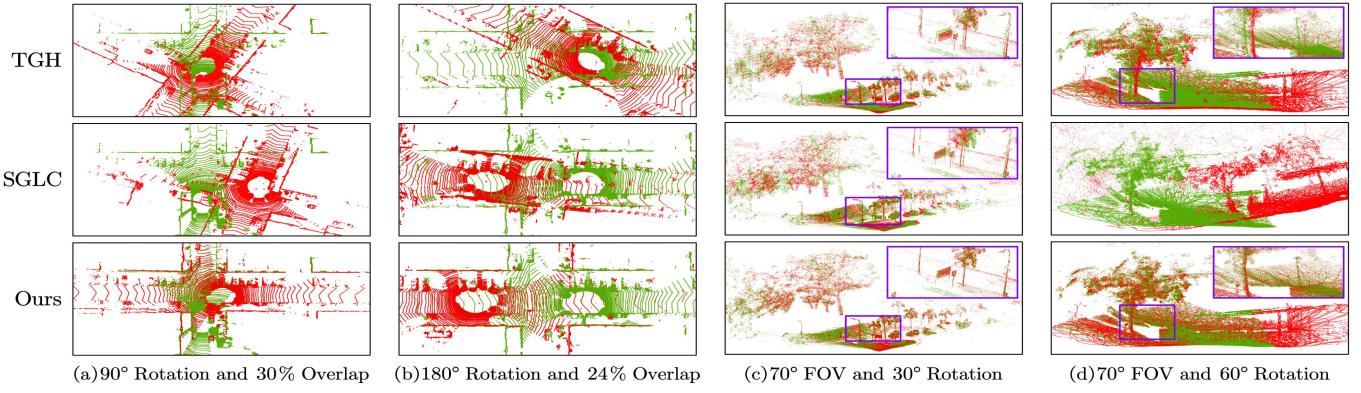


Fig. 6. Visualization of 6-DoF pose estimation under challenging conditions.

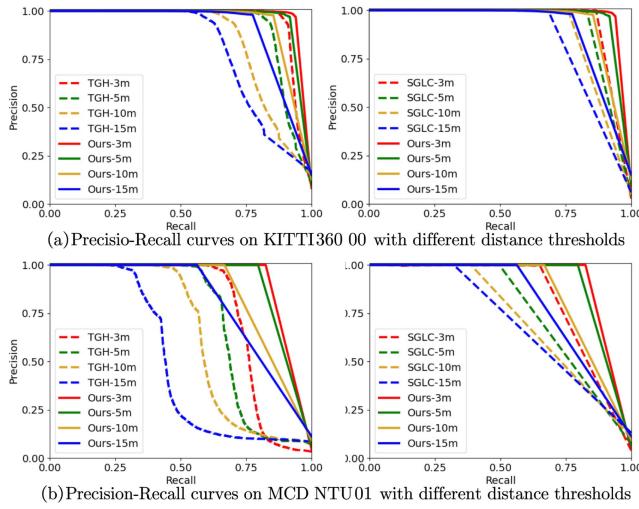


Fig. 7. Precision-Recall curves of ours and other methods under different distance thresholds.

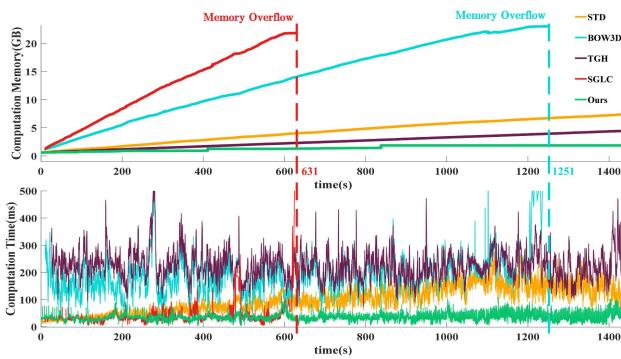


Fig. 8. Time consumption and memory usage of different methods over time.

experiments, we selected STD, BoW3D, TGH, and SGLC. Fig. 8 illustrates the temporal trends in memory usage and computing time for these algorithms, while Table III summarizes their average computing times and final memory usages.

Although SGLC maintains relatively low computing times, it requires extensive feature point cloud storage for geometric verification, resulting in memory overflow at 631 seconds. While

TABLE III
AVERAGE COMPUTING TIMES AND FINAL MEMORY USAGES

	STD	BoW3D	TGH	SGLC	Ours
Time(ms)	94.5	179.2	215.5	34.6	33.3
Memory(%)	31.2	95.9	18.7	90.9	7.8

TABLE IV
ABLATION STUDY ON KITTI00

d_1	d_2	d_3	θ	l	w/o RANSAC	RANSAC
✗	✓	✓	✓	✓	50.1/50.0	98.5/80.9
✓	✗	✓	✓	✓	57.1/50.0	98.7/77.2
✓	✓	✗	✓	✓	85.5/52.0	97.5/69.5
✓	✓	✓	✗	✓	16.2/50.0	46.3/64.3
✓	✓	✓	✓	✗	16.2/50.0	93.2/87.8
✓	✓	✓	✓	✓	99.5/92.9	99.7/99.5

* d_1 , d_2 and d_3 are represent the different dimensions in sub-descriptor. θ and l represent the angle and edge length information, respectively.

TGH ensures lower memory usage, its joint optimization during pose estimation incurs significant computational overhead, hindering real-time performance. In contrast, our method achieves the lowest and most stable time consumption and memory usage. The lightweight descriptors effectively reduce database memory requirements, and the high-quality node matching mechanism enables accurate pose estimation without additional time-consuming geometric verification.

E. Ablation Study

As detailed in III-B, we constructed three-dimensional sub-vector for each semantic configuration based on triangular topological relationships. To validate the rationale behind our descriptor design, we performed ablation experiments to assess the impact of each component on descriptor performance. We calculated the F1 MAX and EP metrics for the algorithm on KITTI00 under various conditions. The experimental results, presented in Table IV, indicate that d_1 , d_2 , and d_3 represent the different dimensions in sub-vector, θ represents angle information, and l denotes edge length information. The results show that three-dimensional descriptors offer superior place

recognition performance compared to two-dimensional descriptors. Although applying RANSAC to remove outlier pairs can enhance the performance, it still lags behind 3D due to reduced matching quality. Additionally, the experiment highlighted the limitations of using only angle and edge length information for descriptor construction. Our method effectively integrates these two types of information, leading to a significant improvement in algorithm performance.

V. CONCLUSION

In this letter, an accurate, efficient semantic graph-based LiDAR loop closure detection method SGT-LCC is presented. Unlike previous works, the proposed method comprehensively considers various semantic and spatial topology in semantic graphs, rather than merely focusing on the adjacency between semantic nodes. We encode the descriptor corresponding to each semantic node in the graph based on the second-order semantic topological configurations and extend the second-order topological relations to triangular spatial topology for extracting geometric sub-descriptor. The proposed descriptor can effectively associate similar semantic graphs while estimating 6-DoF pose between point clouds. In addition, we integrate local descriptors from various nodes in the graph through fuzzy classification, enabling lightweight database and efficient global search. Finally, extensive experiments prove that our method has superior performance and robustness compared to other state-of-art methods. Additionally, our method operates in real time with friendly time consumption and memory usage, making it suitable for integration into semantic SLAM systems. In fact, semantic graph is an effective approach for robotic scene understanding and navigation. In future work, we will explore integrating the proposed method into other higher-level tasks within the robotic system.

REFERENCES

- [1] 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. Automat. Lett.*, vol. 8, no. 5, pp. 2828–2835, May 2023.
- [2] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, “STD: Stable triangle descriptor for 3D place recognition,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1897–1903.
- [3] P. Yin et al., “Outram: One-shot global localization via triangulated scene graph and global outlier pruning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 13717–13723.
- [4] 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.
- [5] L. Li et al., “SSC: Semantic scan context for large-scale place recognition,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2092–2099.
- [6] B. Jiang and S. Shen, “Contour context: Abstract structural distribution for 3D LiDAR loop detection and metric pose estimation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 8386–8392.
- [7] H. Kim, J. Choi, T. Sim, G. Kim, and Y. Cho, “Narrowing your FOV with SOLiD: Spatially organized and lightweight global descriptor for FOV-constrained LiDAR place recognition,” *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9645–9652, Nov. 2024.
- [8] N. Wang, X. Chen, C. Shi, Z. Zheng, H. Yu, and H. Lu, “SGLC: Semantic graph-guided coarse-fine-refine full loop closing for LiDAR SLAM,” *IEEE Robot. Automat. Lett.*, vol. 9, no. 12, pp. 11545–11552, Dec. 2024.
- [9] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, “Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8349–8356, Oct. 2021.
- [10] W. Ma, S. Huang, and Y. Sun, “Triplet-graph: Global metric localization based on semantic triplet graph for autonomous vehicles,” *IEEE Robot. Automat. Lett.*, vol. 9, no. 4, pp. 3155–3162, Apr. 2024.
- [11] M. Liao, Y. Zhang, J. Zhang, L. Liang, S. Coleman, and D. Kerr, “Semantic topological descriptor for loop closure detection within 3D point clouds in outdoor environment,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2856–2863.
- [12] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet ++: Fast and accurate LiDAR semantic segmentation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4213–4220.
- [13] Y. Zhang et al., “Polarnet: An improved grid representation for online LiDAR point clouds semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9598–9607.
- [14] H. Tang et al., “Searching efficient 3D architectures with sparse point-voxel convolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 685–702.
- [15] Y. Cao, Y. Wang, Y. Xue, H. Zhang, and Y. Lao, “FEC: Fast Euclidean clustering for point cloud segmentation,” *Drones*, vol. 6, no. 11, 2022, Art. no. 325.
- [16] J. Behley et al., “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9296–9306.
- [17] 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.
- [18] 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.
- [19] 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. 6382–6391.
- [20] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, “MulRan: Multimodal range dataset for urban place recognition,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6246–6253.
- [21] T. -M. Nguyen et al., “MCD: Diverse large-scale multi-campus dataset for robot perception,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 22304–22313.