

# SGLC: Semantic Graph-Guided Coarse-Fine-Refine Full Loop Closing for LiDAR SLAM

Neng Wang <sup>ID</sup>, Xieyuanli Chen <sup>ID</sup>, Member, IEEE, Chenghao Shi <sup>ID</sup>, Zhiqiang Zheng, Hongshan Yu <sup>ID</sup>, and Huimin Lu <sup>ID</sup>, Member, IEEE

**Abstract**—Loop closing is a crucial component in SLAM that helps eliminate accumulated errors through two main steps: loop detection and loop pose correction. The first step determines whether loop closing should be performed, while the second estimates the 6-DoF pose to correct odometry drift. Current methods mostly focus on developing robust descriptors for loop closure detection, often neglecting loop pose estimation. A few methods that do include pose estimation either suffer from low accuracy or incur high computational costs. To tackle this problem, we introduce SGLC, a real-time semantic graph-guided full loop closing method, with robust loop closure detection and 6-DoF pose estimation capabilities. SGLC takes into account the distinct characteristics of foreground and background points. For foreground instances, it builds a semantic graph that not only abstracts point cloud representation for fast descriptor generation and matching but also guides the subsequent loop verification and initial pose estimation. Background points, meanwhile, are exploited to provide more geometric features for scan-wise descriptor construction and stable planar information for further pose refinement. Loop pose estimation employs a coarse-fine-refine registration scheme that considers the alignment of both instance points and background points, offering high efficiency and accuracy. Extensive experiments on multiple publicly available datasets demonstrate its superiority over state-of-the-art methods. Additionally, we integrate SGLC into a SLAM system, eliminating accumulated errors and improving overall SLAM performance.

**Index Terms**—SLAM, localization.

## I. INTRODUCTION

L OOP closing plays a crucial role in simultaneous localization and mapping (SLAM) systems for correcting odometry drifts and building consistent maps, especially in single-sensor SLAM without GPS information. LiDAR-based loop

Received 6 July 2024; accepted 22 October 2024. Date of publication 11 November 2024; date of current version 18 November 2024. This article was recommended for publication by Associate Editor B. Englot and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation of China under Grant 62403478, in part by the 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. (Corresponding authors: Xieyuanli Chen; Huimin Lu.)

Neng Wang, Xieyuanli Chen, Chenghao Shi, Zhiqiang Zheng, and Huimin Lu are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410008, China (e-mail: xieyuanli.chen@nudt.edu.cn; lhmnew@nudt.edu.cn).

Hongshan Yu is with the Hunan University, Changsha 410012, China.

The implementation of SGLC will be released at <https://github.com/nubot-nudt/SGLC>.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3495455>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3495455

closing typically requires feature extraction from LiDAR scans to generate discriminative descriptors. This task is especially challenging due to the extensive and sparse nature of point clouds in outdoor operations for autonomous robots.

While many existing methods [1], [2], [3], [4], [5], [6] focus on loop closure detection (LCD), few can estimate closed-loop pose in six degrees of freedom (6-DoF). Some methods incorporate pose estimation with 1-DoF [7], [8], [9] or 3-DoF [10], [11], which may not be sufficient for 6-DoF SLAM systems. Existing full 6-DoF pose estimation methods, however, are either very time-consuming [12] or have relatively lower accuracy [13].

To tackle these issues, we propose SGLC, an efficient semantic graph-guided full loop closing framework that offers both robust LCD and accurate 6-DoF loop pose estimation. Different from existing semantic graph-based methods [3], [14] that indiscriminately use both foreground instances and background points, or those overlook the geometrically rich background point clouds [15], our method leverages the distinct properties of both foreground and background elements efficiently. SGLC builds semantic graphs based on foreground instances as they can be naturally represented as individual nodes. While for retrieving loop candidates and estimating 6-DoF poses, SGLC exploits both the topological properties of the semantic graph and the geometric features of background points to enhance loop closing accuracy and robustness.

Specifically, SGLC first generates LiDAR scan descriptors to quickly retrieve multiple candidate scans by exploiting the semantics and typologies of the foreground semantic graph and the geometric features of the background. To prevent incorrect loops from compromising the SLAM system, we apply geometric verification to eliminate false loop candidates by identifying graph node correspondences between the query and candidate scans. During node matching, we ensure accurate correspondences and facilitate the process through an outlier correspondence pruning method based on their neighbor geometric structure. Finally, for the verified loop candidate scan, we propose a coarse-fine-refine registration strategy for estimating 6-DoF pose between it and the query scan. This initially estimates a coarse pose by aligning the sparse matched node centers. Then, a fine registration of dense instance points is applied. Additional planar information from the background points further refines the final pose estimation using point-to-plane constraints. This strategy aligns both foreground instances and background points, with each stage starting from a favorable initial value, ensuring accuracy and efficiency. Fig. 1 illustrates how our method accurately detects loop closures and identifies instance node correspondences, even in places with significant direction and position differences, thus finally estimating the loop pose and closing the loop.

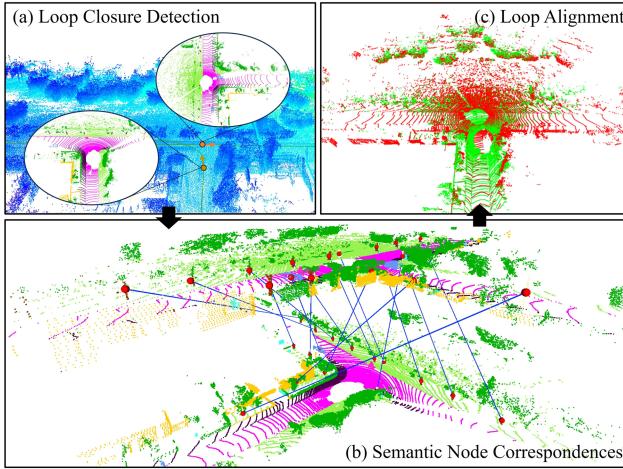


Fig. 1. Visualization of loop closing using our method. (a) Loop closure detection, it shows a reverse loop on the KITTI 08 sequence found by our approach even with significant changes in the position and orientation. (b) Semantic node correspondences for geometric verification and initial loop poses estimation. The blue lines indicate the node correspondences and red spheres represent the estimated instance center. (c) Final alignment for loop correction.

In summary, our contributions are fourfold: (i) We propose a novel semantic graph-guided full loop closing framework, SGLC, that offers robust loop detection and accurate 6-DoF pose estimation. (ii) We design an effective and efficient outlier pruning method to remove incorrect node correspondences. (iii) We proposed a coarse-fine-refine registration scheme that improves the accuracy and efficiency of pose estimation. (iv) We seamlessly incorporate SGLC into a current SLAM framework, reducing the error in odometry estimations. Extensive evaluation results on KITTI [16], KITTI-360 [17], Ford Campus [18] and Apollo [19] datasets support these claims.

## II. RELATED WORK

*Handcrafted-based approaches:* These methods typically utilize the geometric features within the point cloud to describe it. In the early stages, Magnusson et al. [20] explore the application of the Normal Distributions Transform (NDT) for surface representation to develop histogram descriptors based on surface orientation and smoothness. The descriptors perform well in different environments and provide the inspiration for subsequent NDT-based methods [21], [22]. To quickly extract descriptors, some methods project point clouds into 2D plane [4], [7], [8], [10] for encoding features. Among them, Scan Context (SC) family [4], [7], [10] is widely used due to its high efficiency and good LCD performance. It converts the raw point cloud into a polar coordinate bird's eye view (BEV) and encodes the maximum height into the image bins. This yields rotation-invariant ring key descriptors for retrieval and a more detailed 2D image matrix for calculating similarity. Based on that, Jiang et al. [11] employed a similar projection-based approach, but it generates multi-layered BEV images, which can effectively reduce the loss of height information. Recently, Cui et al. [13] propose BoW3D, leveraging Link3D features [23] to build bag of words and estimate 6-DoF pose.

*Deep Neural Networks (DNN)-based approaches:* These methods leverage networks to extract local features and then

aggregate them into a global descriptor for retrieval. The early DNN-based work PointNetVLAD [2] is a combination of PointNet [24] for extracting features and NetVLAD [25] for generating the final descriptor. After this, a variety of learning-based loop closing methods begin to proliferate. Liu et al. [26] proposed an adaptive feature extraction module and a graph-based neighborhood aggregation module for enhancing PointNetVLAD. In contrast to them, Komorowski et al. [27] build a sparse voxelized point cloud representation and extract features by sparse 3D convolutions. Chen et al. [9] propose OverlapNet, a range image-based LCD method by estimating image overlap. A subsequent enhanced version, OverlapTransformer [5] is proposed with superior performance and higher efficiency. Cattaneo et al. [12] proposed LCDNet, a robust LCD and 6-DoF pose estimation network validated in various environments, but its high computational complexity makes real-time running challenging. Recently, Luo et al. [6] proposed a lightweight BEV-based network showing high efficiency and LCD capability.

Some methods also incorporate semantic information to enhance the descriptor distinctiveness. Kong et al. [3], [28] and Zhu et al. [15] attempt to build semantic graphs for loop closing, while Li et al. [10] utilize semantics to improve existing descriptors. Building upon LCDNet [12], Arce et al. [29] integrate semantic information into the network only during the training phase, which improves loop closing performance and makes the model versatile. We agree that semantics are beneficial for loop closing, as they help differentiate objects in the scene. However, few semantic-based methods have proven both efficient and effective for 6-DoF loop closing.

## III. SEMANTIC GRAPH-GUIDED LOOP CLOSING

This section details our semantic graph-guided loop closing approach, dubbed SGLC shown in Fig. 2, which includes building graphs from the raw point clouds (Section III-A), LiDAR scan descriptor generation (Section III-B), geometric verification (Section III-C) and pose estimation (Section III-D).

### A. Semantic Graph

Semantic graph is a fundamental component of our SGLC, which generates distinctive descriptors and guides the subsequent geometric verification and pose estimation. Given a raw LiDAR scan  $\mathcal{S}$ , semantic label for each point can be obtained using an existing semantic segmentation method [30] with the ability to distinguish between moving and static objects. We then apply clustering [31] to such semantic point clouds to identify object instances. Subsequently, the bounding box of instances can be estimated by enclosing these clusters. As foreground instances such as *pole* and *trunk* are naturally standalone, they can be easily represented as individual nodes in the graph. Background point clouds, such as *building* and *fence*, typically have extensive points and are viewpoint-dependent, making them unstable for representation as instance nodes. Therefore, we construct semantic graphs only for stable foreground instances, such as *pole*, *trunk*, *lamp*, *static vehicle*. Each node  $v$  comprises instance center position  $\mathbf{c} = [x, y, z]^T$ , bounding box size  $\mathbf{b} = [l, h, w]$ , a semantic label  $l_v$ , and a node descriptor  $\mathbf{f} \in \mathbb{R}^D$  generated based on the semantic graph typologies.

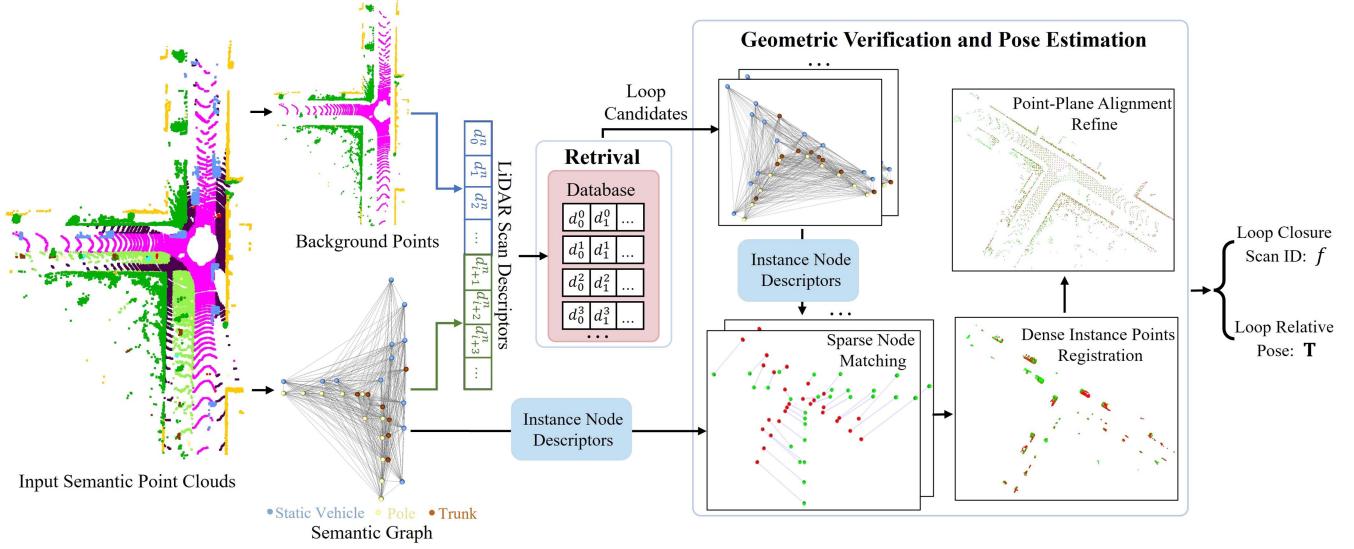


Fig. 2. The framework of SGLC. It first builds a semantic graph for foreground instances and then generates LiDAR scan descriptor considering both the topological properties of the semantic graph and the appearance characteristics of the background. The LiDAR scan descriptor is utilized to retrieve loop candidate scans from the database. Following this, geometric verification is performed on each loop candidate to filter out false loop closure, with the key step utilizing the instance node descriptors for robust sparse node matching. Finally, a coarse-fine-refine registration scheme is employed to estimate the precise 6-DoF pose.

Graph edges are established between pairs of nodes if their spatial Euclidean distance is less than  $d_{\max}$ . Each edge  $e = (\mathbf{v}_i, \mathbf{v}_j)$  is described by a label  $l_e$  determined by the two nodes it connects and the length  $d$ . The label  $l_e$  include categories such as *pole-lamp*, *pole-trunk*, and so on. Finally, we obtain the semantic graph  $\mathcal{G}$  with a set of nodes  $\mathcal{V}$  and a set of edges  $\mathcal{E}$ . The graph adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is created as:

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{E}, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Based on the semantic graph, we design instance node descriptors for the subsequent robust node matching. We first encode the local relationships of nodes by using connected edges. These edges are categorized and quantified to construct a histogram-based descriptor. Specifically, for a node  $\mathbf{v} \in \mathcal{V}$ , we categorize all connected edges into different intervals based on their labels and lengths, and count the occurrences within each interval with length  $d_i$  to generate the descriptor  $\mathbf{f}_l$ .

Since  $\mathbf{f}_l$  only captures the local typologies within the node neighborhood, it does not contain node global properties, such as their global centrality. To enhance the distinctiveness of the descriptor for robust node matching, we employ eigenvalue decomposition on the graph adjacency matrix  $\mathbf{A}$ , yielding  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T$ . The eigenvalues in  $\text{diag}(\Lambda) = \lambda_1, \dots, \lambda_n$  are arranged in descending order. The  $i$ -th row of  $\mathbf{Q}$  represents the  $i$ -th instance node embeddings, which capture the node's global properties in the graph [32]. Therefore, we use the each row of  $\mathbf{Q}$  as part of the instance node descriptor. Considering that the dimensions of the matrix  $\mathbf{Q}$  generated by each graph are different, and to unify dimensions of the descriptor, we encode each node using the first  $k$  columns of  $\mathbf{Q}$ , i.e., the eigenvectors corresponding to the largest  $k$  eigenvalues, thereby generating a row vector  $\mathbf{f}_g \in \mathbb{R}^k$  for each node. Note that the signs of eigenvectors are arbitrary, hence we use their absolute values.

Finally, by concatenating  $\mathbf{f}_l$ ,  $\mathbf{f}_g$ , we obtain each instance node's descriptor  $\mathbf{f} \in \mathbb{R}^D$  in the graph.

### B. LiDAR Scan Descriptor for Retrieval

Based on our semantic graph, our method effectively detects the loop closure. Theoretically, we could use local node descriptors to determine graph similarity and find loops directly. However, graph matching using nodes for LCD on a scan-scan basis is time-consuming, making it unsuitable for real-time SLAM systems. For fast loop candidate retrieving, we design a novel global descriptor for each LiDAR scan, including the foreground descriptor  $\mathbf{F}_f$  and background descriptor  $\mathbf{F}_b$ .

We generate  $\mathbf{F}_f$  based on our proposed foreground semantic graph, using the graph edges and nodes. The first part of  $\mathbf{F}_f$  is similar to the first part of node descriptor  $\mathbf{f}_l$  as an edge-based histogram descriptor. Instead of considering only the edges connected to a particular node, for  $\mathbf{F}_f$ , we account for all edges within the entire graph. These edges are categorized into subintervals in the histogram based on their types and lengths, forming the global edge descriptor. For the second part, We tally the number of nodes with different labels in the semantic graph to effectively describe the node distribution.

Besides  $\mathbf{F}_f$ , which captures the topological relationships between foreground instances, we further design a descriptor to exploit the extensive geometric information in the background point clouds. Inspired by Scan Context [7] encoding the maximum height of the raw point cloud into different bins of a polar BEV image, we leverage semantic information to replace the height data, constructing similar descriptors based on the background point cloud to generate rotation-invariant ring key descriptors. This descriptor efficiently encodes the appearance characteristics of different background point clouds, offering a concise portrayal at a minimal computational cost.

Finally, we perform L2-normalization on  $\mathbf{F}_f$  and  $\mathbf{F}_b$  to enhance the robustness, and concatenate them to form the LiDAR

scan descriptor  $\mathbf{F} \in \mathbb{R}^{D'}$ . This comprehensive descriptor incorporates features from both the foreground semantic graph and the background, thus possessing stronger retrieval capabilities. The LiDAR scan descriptor is utilized to retrieve the multiple similar candidate scans in the database. The similarity between descriptors is computed using the Euclidean distance within the feature space. In real applications, we use the FAISS library [33] to establish the descriptors database and perform fast parallel retrieving.

### C. Geometric Verification

Once loop candidates are obtained, geometric verification is performed on each candidate to determine whether a true loop closure has occurred. The core of geometric verification is to establish node correspondences and estimate an initial relative pose between the query and candidate scans. The similarity between them is then measured by assessing the degree of alignment, including the following four steps:

*Instance Nodes Matching:* For the query scan with semantic graph  $\mathcal{G}_q$  and the target candidate scan with semantic graph  $\mathcal{G}_t$ , we create an affiliation matrix  $\mathbf{I} \in \mathbb{R}^{N \times M}$  based on node descriptor similarity, where  $N$  and  $M$  are the number of nodes in the query and target graphs, respectively. Each element  $\mathbf{I}_{ij}$  is the cost of assigning the  $i$ -th node in  $\mathcal{G}_q$  to the  $j$ -th node in  $\mathcal{G}_t$  as:

$$\mathbf{I}_{ij} = \begin{cases} 1 - \frac{\mathbf{f}_i^q \cdot \mathbf{f}_j^t}{\|\mathbf{f}_i^q\| \times \|\mathbf{f}_j^t\|} & \text{if } l_{vi}^q = l_{vj}^t \text{ and } \mathbf{b}_i^q = \mathbf{b}_j^t, \\ 10^8 & \text{otherwise} \end{cases} \quad (2)$$

A high cost is assigned to reject a match if their node labels are inconsistent or bounding box sizes differ significantly, i.e., the difference in each dimension of the box beyond  $d_b$ .

The matching results are then determined using the Hungarian algorithm [34], yielding the instance node matching correspondence set between  $\mathcal{G}_q$  and  $\mathcal{G}_t$  as:

$$\mathcal{M}_q = \{\mathbf{c}_1^q, \mathbf{c}_2^q, \dots, \mathbf{c}_o^q\}, \quad (3)$$

$$\mathcal{M}_t = \{\mathbf{c}_1^t, \mathbf{c}_2^t, \dots, \mathbf{c}_o^t\}, \quad (4)$$

where  $\mathbf{c} = [x, y, z]^T$  is the position of node mentioned earlier,  $o$  is the number of node matching pairs.

*Node Correspondences Pruning:* Due to changes in viewpoint during revisit or errors in semantic segmentation, there are incorrect node correspondences. Although RANdom SAMple Consensus (RANSAC) [35] can exclude some outliers, using it with all node matches substantially increases the number of iterations, making it unsuitable for online applications.

To reduce the computational complexity, we propose an efficient outlier pruning scheme based on the local structure of nodes before applying RANSAC. We posit that two nodes of a true positive correspondence should exhibit consistent local geometric structures. For a node  $\mathbf{v}$ , its neighboring nodes within a certain range  $d_l$  can be represented as  $\{\mathbf{v}_j\}_{j=1}^K$ , where  $K$  is the total number of neighbors.  $\mathbf{v}$  and any two of its neighbors can form a triangle, denoted as  $\Delta_j$ , and all local triangles of  $\mathbf{v}$  is  $\{\Delta_j\}_{j=1}^{K(K-1)/2}$ . In an ideal scenario, corresponding local geometric triangles of matched nodes should be perfectly aligned. However, practical applications are subject to errors from semantic segmentation and clustering. Therefore, we relax this condition to prevent the elimination of correct correspondences. For the matching of local triangles, we consider two triangles

to be consistent when their corresponding sides are of equal length. Once the number of successfully matched local triangles between two nodes exceeds  $t_m$ , they are considered as true node correspondences.

*Initial Pose Estimation:* Based on pruned correspondences, denoted as  $\mathcal{M}'_q$  and  $\mathcal{M}'_t$ , we estimate a relative pose between the query and candidate scans based on RANSAC and SVD decomposition [36]. In each RANSAC iteration, we randomly select three matching pairs of  $\mathcal{M}'_q$  and  $\mathcal{M}'_t$  to solve the transformation equation based on SVD decomposition as:

$$\mathbf{R}_i, \mathbf{t}_i = \min_{\mathbf{R}, \mathbf{t}} \sum_{j=1}^3 \|\mathbf{R} \cdot \mathbf{c}_j^t + \mathbf{t} - \mathbf{c}_j^q\|_2^2, \quad (5)$$

by which we obtain a transformation in each iteration and finally select the transformation with the greatest number of inliers from all iterations as a coarse pose estimation for loop candidate verification  $\mathbf{T}_{\text{coarse}} = \{\mathbf{R}_c, \mathbf{t}_c\}$ .

*Loop Candidate Verification:* We verify the loop candidates by calculating scan similarity on two levels. First, we evaluate the semantic graph similarity between the query scan and candidate scan as follows:

$$S_{\text{graph}} = \exp \left( - \frac{\sum_{j=1}^u \|\mathbf{R}_c \cdot \mathbf{c}_j^t + \mathbf{t}_c - \mathbf{c}_j^q\|_2}{u} \right), \quad (6)$$

where  $\mathbf{c}_j^t, \mathbf{c}_j^q$  are the inlier correspondences.  $u$  is the number of inliers.  $S_{\text{graph}}$  measures the alignment of the two graphs.

Furthermore, we use  $\mathbf{T}_{\text{coarse}}$  to align the background point clouds of the query and candidate scans, and then calculate the cosine similarity of their background descriptors  $\mathbf{F}_b^q$  and  $\mathbf{F}_b^t$ . This provides greater reliability compared to using graph similarity alone. If both  $S_{\text{graph}}$  and the background similarity exceed thresholds  $t_g$  and  $t_b$ , the query scan and the candidate scan are considered a true loop closure. We then proceed to estimate their precise pose transformation as follows.

### D. 6-DoF Pose Refinement

The aforementioned  $\mathbf{T}_{\text{coarse}}$ , estimated at the sparse instance node level, may not be accurate and robust enough for closing the loop. To enhance the accuracy of the relative pose estimation, we propose performing a dense points registration on all foreground instance points from the input semantic point cloud, using  $\mathbf{T}_{\text{coarse}}$  as the initial transformation.

Benefiting from the already obtained instance node matches, finding the corresponding point matches between matched instances becomes easy and fast due to the significantly reduced search space. In fact,  $\mathbf{T}_{\text{coarse}}$  already initially align the query scan and candidate scan. Therefore, directly searching for the nearest neighbor points as point matches is sufficient to initialize and solve the Iterative Closest Point (ICP) as:

$$\mathbf{T}_{\text{icp}} = \min_{\mathbf{T}} \sum_{(\mathbf{p}_j^q, \mathbf{p}_j^t) \in \mathcal{C}} \|\mathbf{T} \mathbf{p}_j^t - \mathbf{p}_j^q\|_2^2, \quad (7)$$

where  $\mathcal{C}$  is the set of instance points nearest neighbor correspondences between query scan and loop scan.

The dense registration result  $\mathbf{T}_{\text{icp}}$  encompasses the registration of all foreground instance points. However, background points, such as those from buildings and roads offering stable plane features, are also valuable for pose estimation. Therefore,

TABLE I  
THE PARAMETERS SETTINGS OF SGLC

Parameter	Value	Description
$d_{\max}$	60m	edge threshold for semantic graph
$d_i$	2m	interval for edge histogram descriptors
$k$	30	the dimension for eigenvectors
$d_b$	2m	difference threshold of bounding box dimension
$d_l$	20m	the range for generating local triangle
$t_m$	2	threshold for determining true node correspondences
$t_g$	0.5	threshold for semantic graph similarity
$t_b$	0.5	threshold for background similarity
$D$	120	the dimension for node descriptor
$D'$	231	the dimension for LiDAR scan descriptor

to further optimize the pose accuracy from dense points registration, we utilize point-to-plane residuals to refine the final relative pose.

$$\mathbf{T}_{\text{refine}} = \min_{\mathbf{T}} \sum_{(\mathbf{p}_j^{q'}, \mathbf{p}_j^{t'}) \in \mathcal{C}'} \|\mathbf{n}_j^T (\mathbf{T}\mathbf{p}_j^{t'} - \mathbf{p}_j^{q'})\|_2^2, \quad (8)$$

where  $\mathcal{C}'$  is the set of point correspondences with consistent plane normal vector, denoted as  $\mathbf{n}_j^T \in \mathbb{R}^3$ .

$\mathbf{T}_{\text{coarse}}$  is estimated via sparse node matching, while  $\mathbf{T}_{\text{icp}}$  is determined by dense instance points registration. This sparse-to-dense registration mechanism enhances pose estimation accuracy in a coarse-to-fine manner. Finally, the pose is further refined using point-to-plane constraints for background points. This semantically-guided coarse-fine-refine pose estimation process considers the alignment of both instance points and background points. By separately handling instance points and background points, each alignment stage begins with a favorable initial value, ensuring fast matching and convergence, as well as multi-step checked robust pose estimation.

#### IV. EXPERIMENTAL EVALUATION

##### A. Experimental Setup

**Dataset:** We evaluate our method on KITTI [16] and KITTI-360 [17] datasets. Additionally, Ford Campus [18] and Apollo [19] datasets are used to test its generalization ability.

**Implementation Details:** The parameters used in our experiments are listed in Table I. Due to background similarity being easily affected by semantic segmentation quality, we set  $t_b$  for different datasets ( $t_b = 0.5$  for KITTI, KITTI-360 and Apollo,  $t_b = 0.45$  for Ford Campus). Other parameters remain unchanged. We use the semantic information from a semantic segmentation network [30] on all datasets and we train different models on the KITTI dataset to ensure that each test sequence remains unseen during the training phase. We mainly build semantic graphs from *static vehicle*, *pole* and *trunk* instances, and utilize background points from *building*, *fence*, *road* and *vegetation*. This is convenient for replacing or adding other semantic information. To fairly compare all semantic-assisted loop closing methods, we evaluate them using the same labels from SegNet4D [30]. All the experiments are conducted on a machine with an AMD 3960X @3.8 GHz CPU and a NVIDIA RTX 3090 GPU. More implementation details can be found in our online supplementary.<sup>1</sup>

<sup>1</sup><https://github.com/nubot-nudt/SGLC/Supplementary.pdf>

##### B. Loop Closure Detection

We follow the experimental setups of Li et al. [10] to evaluate LCD performance and regard LiDAR scan pairs as positive samples of loop closure when their Euclidean distance is less than 3 m, as negative samples if the distance exceeds 20 m. For the KITTI dataset, we perform performance comparisons on all sequences with loop closure. For the KITTI-360 dataset, in line with [12], we focus on sequences 0002 and 0009 with the highest number of loop closures. Besides, we also report the results on the Ford Campus dataset (sequence 00) and a subset of the Apollo Columbia Park MapData proposed by [37] for generalization evaluation.

**Metrics:** Following [10], [13], we use the maximum  $F_1$  score and Extended Precision ( $EP$ ) as evaluation metric.

**Baselines:** We compare the results with SOTA baselines, including DNN-based methods: PointNetVLAD (PV) [2], SGPR [3], LCDNet [12], OverlapTransformer (OT) [5], BEVPlace [6], PADLoC [29], as well as handcrafted-based methods: Scan Context (SC) [7], GOSMatch [15], SSC [10], BoW3D [13], Contour Context (CC) [11].

**Results:** The results are shown in Table II. Our method outperforms SOTA on multiple sequences and achieves the best average  $F1\text{max}$  score and  $EP$  for the KITTI dataset. Specifically, for sequence 08 with many reverse loop closures, our approach still exhibits superior performance, proving its good rotational invariance. And on the KITTI360 dataset, our method remains competitive. Additionally, we report the average runtime for all methods, including descriptor generation and loop closure determination. Our method performs as fast as handcrafted methods while achieving superior performance in most cases, demonstrating both efficiency and effectiveness.

Furthermore, to investigate the loop closure detection performance at further distances, we followed [5], [19] regarding two LiDAR scans as a loop closure when their overlap ratio is beyond 0.3, which indicates that the maximum possible distance between them is around 15 m. And we adopt the same experimental setup as theirs to evaluate our method on the KITTI 00 sequence using AUC,  $F1\text{max}$ , Recall@1, and Recall@1% as metrics. The results are shown in Table III. Due to BoW3D leaning more towards geometric verification, we are unable to generate its AUC and recall@1% results from its open-source implementation. From the results, our method significantly outperforms SOTA baselines in terms of overall metric, indicating its robustness.

To evaluate the generalization capability of SGLC, we conducted additional experiments on the Ford Campus and Apollo datasets, focusing on loop pairs with an overlap ratio greater than 0.3. We primarily compared semantic-assisted methods to assess their adaptability to scene semantic changes or label quality declines. Due to the page limitation, we show the semantic segmentation details in our online supplementary. As shown in the Table IV, our method demonstrates the best performance on both the Ford Campus and Apollo datasets, highlighting its strong generalization capability. When the quality of semantic segmentation declines, some instances may be partially segmented, but after clustering, they still form a single object, which does not affect the construction of our semantic graph. Once the semantic graph is built, the LiDAR scan descriptors retrieve multiple candidate scans, including the correct loop scan. Through our geometric verification via graph matching, the true loop closure is identified. Besides, for short-term loop

TABLE II  
THE PERFORMANCE COMPARISON OF F1 MAX SCORES AND EXTENDED PRECISION ON THE KITTI AND KITTI-360 DATASETS

Methods	KITTI							KITTI-360		Time (ms)
	00	02	05	06	07	08	Mean	0002	0009	
DNN-based	PV	0.779/0.641	0.727/0.691	0.541/0.536	0.852/0.767	0.631/0.591	0.037/0.500	0.595/0.621	0.349/0.515	0.330/0.510
	SGPR*	0.720/0.507	0.823/0.531	0.720/0.552	0.680/0.524	0.700/0.500	0.683/0.506	0.721/0.520	0.862/0.506	0.845/0.503
	LCDNet	0.970/0.847	<b>0.966/0.917</b>	<b>0.969/0.938</b>	0.958/0.920	0.916/0.684	<b>0.989/0.908</b>	0.961/0.869	<b>0.998/0.993</b>	0.988/0.734
	OT	0.873/0.800	0.810/0.725	0.837/0.772	0.876/0.809	0.625/0.505	0.667/0.510	0.781/0.687	0.796/0.507	0.879/0.528
	BEVPlace	0.960/0.849	0.845/0.819	0.885/0.815	0.895/0.815	0.917/0.687	0.967/0.868	0.912/0.809	0.920/0.504	0.938/0.684
Handcrafted	PADLoC*	0.983/0.912	0.920/0.880	0.957/0.863	0.981/0.944	0.781/0.704	0.910/0.718	0.922/0.837	0.948/0.596	0.969/0.797
	SC	0.750/0.609	0.782/0.632	0.895/0.797	0.968/0.924	0.662/0.554	0.607/0.569	0.777/0.681	0.771/0.554	0.851/0.619
	GOSMatch*	0.916/0.535	0.694/0.575	0.785/0.611	0.491/0.518	0.947/0.913	0.908/0.571	0.790/0.621	0.694/0.500	0.766/0.575
	SSC*	0.955/0.865	<u>0.933/0.768</u>	<u>0.966/0.956</u>	<u>0.983/0.978</u>	0.894/0.785	0.950/0.940	0.947/0.882	<u>0.965/0.793</u>	<u>0.966/0.873</u>
	BoW3D	0.977/0.981	0.578/0.704	<b>0.965/0.969</b>	<u>0.985/0.985</u>	0.906/0.929	0.900/0.866	0.885/0.906	0.210/0.560	0.682/0.761
Ours*	CC	0.977/0.964	0.903/0.568	0.958/0.901	<u>0.993/0.959</u>	0.906/0.811	0.823/0.575	0.927/0.796	0.717/0.560	0.886/0.620
	Ours*	<b>0.998/0.986</b>	0.888/0.899	<b>0.969/0.967</b>	<b>0.995/0.963</b>	<b>0.993/0.991</b>	<b>0.988/0.980</b>	<b>0.972/0.964</b>	0.932/0.934	<b>0.994/0.978</b>
[ $F_1$ max / EP ], the best results are highlighted in bold and the second best are underlined. * denotes semantic-assisted method.										
10.2										

[  $F_1$  max / EP ], the best results are highlighted in bold and the second best are underlined. \* denotes semantic-assisted method.

TABLE III  
THE PERFORMANCE EVALUATION OF LOOP CLOSURE DETECTION ON THE KITTI DATASET USING OVERLAP-BASED LOOP PAIRS

Methods	AUC	F1max	Recall @1	Recall @1%
	0.856	0.846	0.776	0.845
DNN-based	SGPR*	0.591	0.575	0.753
	LCDNet	0.933	0.883	0.915
	OT	0.907	0.877	0.906
	BEVPlace	0.926	0.889	0.913
	PADLoC*	0.934	0.903	0.930
	Ours*	<b>0.949</b>	<b>0.931</b>	<b>0.950</b>
0.986				

TABLE IV  
THE GENERALIZATION PERFORMANCE EVALUATION

Methods	Ford Campus				Apollo			
	AUC	F1max	Recall @1	Recall @1%	AUC	F1max	Recall @1	Recall @1%
SGPR	0.412	0.439	0.467	0.951	0.640	0.451	0.626	0.936
GOSMatch	0.752	0.632	0.820	0.954	0.677	0.568	0.739	0.934
SSC	0.924	0.865	<u>0.915</u>	0.964	0.937	<u>0.916</u>	0.917	0.943
PADLoC	0.938	0.873	0.910	0.963	0.721	0.609	0.735	0.910
Ours†	<b>0.955</b>	<b>0.908</b>	<b>0.920</b>	<b>0.970</b>	0.956	0.910	0.933	0.951
Ours	<b>0.959</b>	<u>0.897</u>	0.896	0.960	<u>0.957</u>	<b>0.919</b>	<b>0.956</b>	<b>0.974</b>

† denotes build semantic graph without vehicle node. The best results are highlighted in bold and the second best are underlined.

closing in SLAM, static vehicles can serve as valuable feature nodes to enhance loop detection.

### C. Loop Pose Estimation

To validate the loop pose estimation performance, we follow Cattaneo et al. [12] and evaluate our approach on the KITTI sequences 00 and 08 and KITTI-360 sequences 0002 and 0009. To keep consistent with them, we choose the loop closure samples when the distance is within 4 m. Additionally, we also select the more challenging loop pairs with overlap rate beyond 0.3 for testing.

*Metrics.* (i) Registration Recall (RR), represents the percentage of successful alignment. (ii) Relative Translation Error (RTE), the Euclidean distance between the ground truth and estimated poses. (iii) Relative Rotation Error (RRE), the rotation angle difference between the ground truth and estimated poses. Two scans are regarded as a successful registration when the RTE and RRE are below 2 m and 5°, respectively.

*Baselines.* The baseline methods for comparison mainly include 6-DoF pose estimation methods BoW3D [13] and LCDNet [12]. LCDNet is available in two version: LCDNet (fast) utilizes an unbalanced optimal transport (UOT)-based head to estimate relative pose while LCDNet replaces the head by a RANSAC estimator.

*Results.* As shown on the Table V, Our method achieves the best loop pose estimation performance on both distance-based loop pairs and more challenging overlap-based loop pairs. Initially, both LCDNet and our method achieve a 100% RR for distance-based loop pairs. However, for overlap-based loop pairs, LCDNet's performance declined, while our method still maintains superior registration capabilities, demonstrating its robustness for viewpoint changes. In terms of alignment accuracy, our method also significantly outperforms other baselines. We also present the qualitative results in Fig. 3. As can be seen, our method achieves better alignment even in low overlap loop closure, attributing to the semantically-guided coarse-fine-refine point cloud alignment process. Table VI shows the results on the KITTI-360 dataset with distance-based close loop. It can be observed that LCDNet exhibits the best generalization ability, but our method remains competitive and has the best alignment accuracy, considering both RTE and RRE. We also report the average running time of pose estimation on our machine. From the comparison, our method boasts the fastest running speed. The total runtime of our system, incorporating semantic segmentation, loop closure detection and poses estimation, is 98.1 ms, less than the data acquisition time of 100 ms for a typical rotational LiDAR sensor, thereby rendering it suitable for real-time SLAM systems.

### D. Performance on SLAM System

We integrate SGLC into A-LOAM odometry to eliminate cumulative errors for evaluating the accuracy of SLAM trajectories, and compare with baseline integrated with BoW3D. We utilize a Incremental Smoothing and Mapping (iSAM2) [38]

TABLE V  
THE COMPARISON OF LOOP POSE ESTIMATION ERRORS ON THE KITTI DATASET

	Distance-based						Overlap-based						Time(ms)
	RR(%)	RTE(m)	RRE( $^{\circ}$ )	RR(%)	RTE(m)	RRE( $^{\circ}$ )	RR(%)	RTE(m)	RRE( $^{\circ}$ )	RR(%)	RTE(m)	RRE( $^{\circ}$ )	
BoW3D	95.61	<u>0.06</u>	0.81	77.29	<u>0.09</u>	2.06	57.69	<u>0.07</u>	0.92	46.10	<u>0.10</u>	1.95	40.0
LCDNet(fast)	97.44	0.52	0.60	78.28	0.99	1.29	66.70	0.56	1.02	47.20	0.88	1.29	81.7
LCDNet	<b>100</b>	0.13	0.44	<b>100</b>	0.20	0.57	<u>93.51</u>	0.21	0.81	<u>95.39</u>	0.31	0.94	429.1
Ours	<b>100</b>	<b>0.04</b>	<b>0.21</b>	<b>100</b>	<b>0.08</b>	<b>0.41</b>	<b>99.82</b>	<b>0.04</b>	<b>0.24</b>	<b>99.57</b>	<b>0.08</b>	<b>0.46</b>	<b>20.8</b>

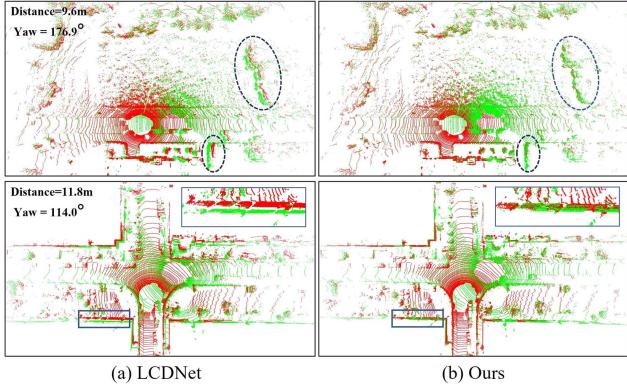


Fig. 3. The qualitative comparison of loop pose estimation on the KITTI dataset using overlap-based loop pairs. Dashed ellipses are directly annotated on the registration results, while solid boxes indicate local magnification. The top-left corner displays the ground truth distance and yaw angle difference of the loop pair.

TABLE VI  
LOOP POSE ESTIMATION RESULTS ON THE KITTI-360 DATASET

	Seq.0002			Seq.0009			Time(ms)
	RR(%)	RTE(m)	RRE( $^{\circ}$ )	RR(%)	RTE(m)	RRE( $^{\circ}$ )	
BoW3D	67.73	<b>0.20</b>	1.99	87.32	0.13	1.31	
LCDNet(fast)	82.92	0.84	1.17	98.81	0.42	1.07	
LCDNet	<b>98.56</b>	0.28	<u>1.03</u>	<b>100</b>	0.19	<u>0.74</u>	
Ours	<u>95.01</u>	<u>0.21</u>	<b>0.89</b>	<u>99.40</u>	<b>0.11</b>	<b>0.51</b>	

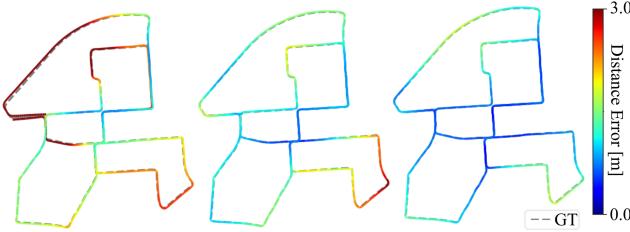


Fig. 4. The trajectory of A-LOAM odometry (left) compared to with integrating BoW3D loop closing method (middle) and integrating our approach (right) on the KITTI 00 sequence.

based pose-graph optimization (PGO)<sup>2</sup> framework to build a factor graph for managing the keyframe poses. If a loop closure is detected, we will add this loop closure constraint into the factor graph to eliminate errors and ensure global trajectories consistency. As shown in Fig. 4, we present the trajectory error analysis between keyframe and ground truth poses. From the

<sup>2</sup><https://github.com/gisbi-kim/SC-A-LOAM>

TABLE VII  
ABLATION STUDIES

	Node Glo.	Out. Pru.	Den. ICP	P-P Ali.	F1max	RR(%)	RTE(m)	RRE( $^{\circ}$ )	Time (ms)
[A]					0.989	97.30	0.21	1.23	9.1
[B]	✓				0.993	97.76	0.20	1.17	+0.5
[C]	✓	✓			<b>0.994</b>	98.29	0.18	0.99	+1.2
[D]	✓	✓	✓		-	99.98	0.07	0.38	+11.8
[E]	✓	✓		✓	-	99.71	<b>0.06</b>	0.34	+8.5
[F]	✓	✓	✓	✓	-	<b>100</b>	<b>0.06</b>	<b>0.31</b>	+20.2
[G]	✓	✓	•		-	<b>100</b>	<b>0.06</b>	0.33	+86.1
[H]	✓	✓	•		-	99.65	<b>0.06</b>	0.34	+9.4

+ denotes the time increment compared to [A]. • denotes direct point-to-point ICP or point-to-plane alignment for the raw scan.

"Node Glo." denotes the global properties of instance nodes. "Out. Pru." denotes outlier pruning for node matches. "Den. ICP" denotes dense points registration, i.e.,  $T_{icp}$ . "P-P Ali." denotes point-to-plane alignment, i.e.,  $T_{refine}$ .

results, our method significantly enhances the SLAM system by reducing the cumulative odometry error, and shows superior performance compared to the baseline that incorporates BoW3D. This improvement is attributed to the excellent loop closing capability of SGLC.

### E. Ablation Study

In this section, we conduct a series of ablation studies on KITTI sequence 00 and 08 and report the mean metrics to evaluate the effectiveness of designed components, and analyze their runtime. The results are presented in Table VII. [A] is our baseline by removing those components listed in Table VII. Comparing [A] and [B], it is demonstrated that incorporating the global properties of instance nodes can enhance the performance of loop closing. As anticipated, it increases each node's distinctiveness, which is beneficial for finding the correct node correspondences. For [C], we can see the effectiveness of outlier pruning in improving registration accuracy with the same number of RANSAC iterations as [B]. In experiment [D], [E] and [F], dense points registration and point-to-plane alignment are applied mainly for loop scan determined after geometric verification, so they do not affect the results of loop closure detection. The results clearly show that both dense points registration and point-to-plane alignment can improve registration performance, and we can get optimal results only when they operate in conjunction. The comparison of [F], [G], [H] also indicates that even direct point-to-point ICP or point-to-plane alignment on the raw scan cannot get optimal results and significantly increase computational costs on direct point-to-point ICP, which demonstrates that our registration process effectively utilizes foreground and background. Besides, each component in our framework exhibits high execution efficiency.

## V. CONCLUSION

This letter presents a semantic graph-guided approach for LiDAR-based loop closing. It builds a semantic graph of the foreground instances for descriptor generation and guides subsequent geometric verification and pose estimation. The designed LiDAR scan descriptor leverages both the semantic graph's topological properties and the appearance characteristics of the background points, thereby enhancing its robustness in LCD. For pose estimation, we proposed a coarse-fine-refine registration scheme that considers the alignment of both instance and background points, offering high efficiency and accuracy. The method supports real-time online operation and is convenient for integration into SLAM systems. The experimental results on multiple datasets demonstrate its superiority.

## REFERENCES

- [1] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 231–237.
- [2] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4470–4479.
- [3] X. Kong et al., "Semantic graph based place recognition for 3D point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8216–8223.
- [4] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2095–2101.
- [5] 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. Automat. Lett.*, vol. 7, no. 3, pp. 6958–6965, Jul. 2022.
- [6] L. Luo et al., "BEVPlace: Learning LiDAR-based place recognition using Bird's eye view images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 8700–8709.
- [7] G. Kim, S. Choi, and A. Kim, "Scan context : Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, Jun. 2022.
- [8] Y. Wang, Z. Sun, C. Xu, S. Sarma, J. Yang, and H. Kong, "LiDAR Iris for loop-closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5769–5775.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] Y. Cui, Y. Zhang, X. Chen, 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.
- [14] Z. Qiao, Z. Yu, H. Yin, and S. Shen, "Pyramid semantic graph-based global point cloud registration with low overlap," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 11202–11209.
- [15] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5151–5157.
- [16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," in *Proc. Int. J. Robot. Res.*, 2013, vol. 32, pp. 1231–1237.
- [17] 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.
- [18] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford Campus vision and LiDAR data set," in *Proc. Int. J. Robot. Res.*, 2011, vol. 30, pp. 1543–1552.
- [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] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, "Appearance-based loop detection from 3D laser data using the normal distributions transform," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 23–28.
- [21] A. Zaganidis, A. Zerntev, T. Duckett, and G. Cielniak, "Semantically assisted loop closure in SLAM using NDT histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4562–4568.
- [22] Z. Zhou et al., "NDT-Transformer: Large-scale 3D point cloud localisation using the normal distribution transform representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5654–5660.
- [23] Y. Cui, Y. Zhang, J. Dong, H. Sun, X. Chen, and F. Zhu, "LinK3D: Linear keypoints representation for 3D LiDAR point cloud," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2128–2135, Mar. 2024.
- [24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [25] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.
- [26] Z. Liu et al., "LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2831–2840.
- [27] J. Komorowski, "MinkLoc3D: Point cloud based large-scale place recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1790–1799.
- [28] L. Li et al., "SA-LOAM: Semantic-aided LiDAR SLAM with loop closure," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7627–7634.
- [29] J. Arce, N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "PADLoC: LiDAR-based deep loop closure detection and registration using panoptic attention," *IEEE Robot. Automat. Lett.*, vol. 8, no. 3, pp. 1319–1326, Mar. 2023.
- [30] N. Wang et al., "SegNet4D: Effective and efficient 4D LiDAR semantic segmentation in autonomous driving environments," 2024, *arXiv:2406.16279*.
- [31] S. Park, S. Wang, H. Lim, and U. Kang, "Curved-voxel clustering for accurate segmentation of 3D LiDAR point clouds with real-time performance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6459–6464.
- [32] G. Nikolentzos, P. Meladianos, and M. Vazirgiannis, "Matching node embeddings for graph similarity," in *Proc. Conf. Advancements Artif. Intell.*, 2017, pp. 2429–2435.
- [33] 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.
- [34] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics*, vol. 2, no. 1/2, pp. 83–97, 1955.
- [35] M. A. Fischler and R. C. 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.
- [36] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [37] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6107–6114, Jul. 2022.
- [38] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," in *Proc. Int. J. Robot. Res.*, 2012, vol. 31, pp. 216–235.