PADLoC: LiDAR-Based Deep Loop Closure Detection and Registration Using Panoptic Attention

José Arce[1], Niclas Vödisch[1], Daniele Cattaneo[1], Wolfram Burgard[2], and Abhinav Valada[1]

***Abstract*—A key component of graph-based SLAM systems is the ability to detect loop closures in a trajectory to reduce the drift accumulated over time from the odometry. Most LiDAR-based methods achieve this goal by using only the geometric information, disregarding the semantics of the scene. In this work, we introduce PADLoC for joint loop closure detection and registration in LiDAR-based SLAM frameworks. We propose a novel transformer-based head for point cloud matching and registration, and to leverage panoptic information during training time. In particular, we propose a novel loss function that reframes the matching problem as a classification task for the semantic labels and as a graph connectivity assignment for the instance labels. During inference, PADLoC does not require panoptic annotations, making it more versatile than other methods. Additionally, we show that using two shared matching and registration heads with their source and target inputs swapped increases the overall performance by enforcing forward-backward consistency. We perform extensive evaluations of PADLoC on multiple real-world datasets demonstrating that it achieves state-of-the-art results. The code of our work is publicly available at http://padloc.cs.uni-freiburg.de.**

***Index Terms*—SLAM, Deep Learning Methods, Loop Closure Detection, Point Cloud Registration, LiDAR**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) is a core task of autonomous mobile robots. Typically, SLAM approaches consist of two steps: alignment of consecutive measurements, e.g., from wheel odometry, followed by loop closure detection and registration. Reliable loop closure detection enables a robot to recognize places it has seen before to optimize its world representation and belief of its current position, reducing the drift over time. Thus, it is considered a fundamental component of SLAM systems. Many SLAM systems have been proposed for different sensor modalities including cameras [1] and LiDARs [2]. While vision-based methods fail in challenging lighting conditions such as illumination changes, LiDAR-based approaches are more robust to such alterations and provide a more accurate representation of the environment. In this work, we address the joint problem of loop closure detection and map registration for LiDAR-based SLAM. A high-level overview of our approach is depicted in Fig. 1.

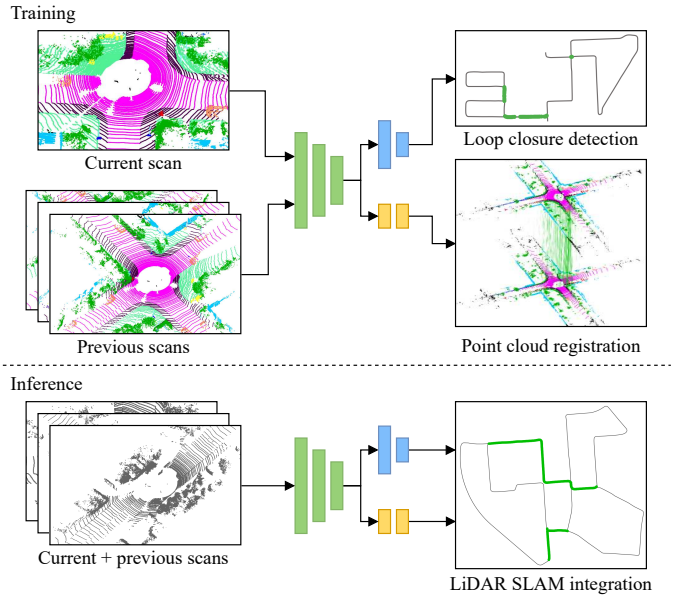


Fig. 1. We propose PADLoC for joint loop closure detection (green areas on the map) and point cloud registration in LiDAR-based SLAM. In addition to geometric information, we leverage panoptic segmentation annotations during training to facilitate more robust point matching. During inference, PADLoC does not require any panoptic information.

Similar to other fields, learning-based approaches have started to replace handcrafted methods [3], [4]. Typically, deep neural networks predict point correspondences which are then used in differential singular value decomposition (SVD) to compute the transformation between two point clouds [5], [6]. Motivated by the success of transformers in natural language processing and computer vision tasks, attention-based architectures were recently introduced for point cloud registration [6], [7], [8] to encode context across points. While existing works do not consider the semantic meaning of the different inputs to a transformer cell, i.e., queries, keys, and values, we explicitly take advantage of the internal structure by feeding in abstract features and raw points separately.

Although geometric information suffices for classical point cloud registration such as Iterative Closest Point (ICP) [9], they can be further stabilized by integrating semantic information [2], [10], [11]. Inspired by recent semantic mapping approaches [10], [12] and methods that exploit panoptic information for vision-based loop closure detection [13], we leverage panoptic segmentation of point clouds in this work. Unlike related methods, our approach requires panoptic labels only while training but not during deployment, making it more versatile. We evaluate the loop closure detection and point cloud

This work was funded by the European Union's Horizon 2020 research and innovation program under grant agreement No 871449-OpenDR and the DFG Emmy Noether Program.
[1] José Arce, Niclas Vödisch, Daniele Cattaneo, and Abhinav Valada are with the Department of Computer Science, University of Freiburg, Germany.
[2] Wolfram Burgard is with the Department of Engineering, University of Technology Nuremberg, Germany.
Digital Object Identifier (DOI): 10.1109/LRA.2023.3239312

registration performance on three real-world autonomous driving datasets, namely, KITTI [14], Ford campus [15], and an in-house dataset recorded in Freiburg, Germany. We compare against both state-of-the-art handcrafted and deep learning-based methods and demonstrate that PADLoC achieves state-of-the-art performance. We also present several ablation studies on the different components of our approach validating our architectural design choices.

The main contributions of this work are as follows:

1) We propose PADLoC, a transformer encoder architecture for point cloud matching and registration. Unlike existing methods, we use separate inputs as keys, values, and queries effectively, exploiting the transformer structure.
2) We define a novel loss function that leverages panoptic information for registration. We further propose formulating both geometric and panoptic registration losses as bidirectional functions that greatly improve performance.
3) We study the effect of multiple weighting methods in SVD to enhance point matching.
4) We extensively evaluate our proposed approach and compare it to other point cloud matching and registration methods, using two openly available datasets and in-house data recorded in Freiburg, Germany.
5) We release our code and the trained models at http://padloc.cs.uni-freiburg.de.

II. Related Work

In this section, we first provide an overview of LiDAR-based loop closure detection techniques for SLAM, followed by various methods for point cloud registration, and finally describe approaches that leverage semantic segmentation for either task.

Loop Closure Detection: Traditionally, handcrafted methods for LiDAR loop closure detection can be categorized into local feature-based and global feature-based methods. Inspired by the success of local feature-based methods in images, approaches from the first category design similar descriptors and adapt them to 3D point cloud data. 3D keypoint descriptors such as Fast Point Feature Histograms (FPFH) [16] and Normal-Aligned Radial Features (NARF) [17] are used to extract local features, which are then aggregated in a bag-of-word model to detect loop closures. More recently, HOPN [18] exploits a bird's-eye view (BEV) representation and normal information to increase robustness to noise and viewpoint changes. Global feature-based approaches, on the other hand, summarize the whole point cloud into a single fingerprint, which is then compared against the fingerprints from past frames to detect loops. The M2DP [19] descriptor projects the point cloud into multiple 2D planes and combines density information computed on each plane into a global descriptor. Scan Context [20] combines a polar coordinate representation with partitioning to generate an image as a global descriptor. Subsequent works extended this method by adding additional information such as intensity [21] and semantic data [22]. Recently, many deep learning-based approaches have been proposed to overcome some of the limitations of handcrafted methods. PointNetVLAD [23] is built on top of the PointNet [24] architecture and generates a compact descriptor. OverlapNet [25] projects the point cloud into a range image and predicts the overlap and the yaw misalignment between a pair of frames. To increase viewpoint robustness and to reduce inference time, OverlapTransformer [26] adapts OverlapNet by including a transformer module. In this work, we build upon LCDNet [5] that uses learning-based feature extraction to generate global descriptors. LCDNet significantly improves loop closure in challenging conditions, such as reverse loops and, unlike other methods, does not require an ad-hoc function to compare two global descriptors.

Point Cloud Registration: Standard techniques for point cloud registration can be broadly classified into two main categories. The first category comprises the Iterative Closest Point (ICP) algorithm [9] and its variants [10], [27]. These methods require an initial guess on the transformation and then iteratively alternate between finding matches between points by exploiting some heuristics and estimating the transformation based on these matches. Methods of the second category use a two-stage approach. They first extract local point features, e.g., FPFH [16], and then regress the transformation using robust estimators such as RANSAC [28]. While methods of the first category are prone to get stuck in local minima if the provided initial guess is not accurate enough, approaches of the second category are sensitive to noise and incorrect matches. Many deep learning-based approaches have also been proposed to solve the point cloud registration task. PointNetLK [29] is a pioneering work that combines an architecture inspired by PointNet [24] and a modified Lucas-Kanade algorithm to iteratively improve the registration. Inspired by the success of transformers in other fields, Deep Closest Point [6] uses an attention-based module to predict soft matches between two point clouds, which are fed to a differentiable SVD layer to infer a rigid transformation. Following the same idea, both GeoTransformer [7] and REGTR [8] directly learn to predict point correspondences using both self and cross-attention. Our previous work LCDNet [5] combines a state-of-the-art feature extraction architecture with a place recognition head and a relative pose head for simultaneous loop closure detection and point cloud registration. In this work, we adapt LCDNet [5] by integrating a transformer-based registration and matching module.

Semantic-Aided Mapping and Localization: Only a handful of works have proposed to leverage semantic information for large-scale mapping and localization [10], [30], and particularly for loop closure detection. Based on semantic segmentation, SuMa++ [10] filters dynamic objects from a LiDAR-based map and extends the ICP algorithm with additional semantic constraints. While SuMa++ does not utilize semantic information for loop closure detection, RINet [31] explicitly addresses LiDAR-based place recognition via a rotation invariant global descriptor combining semantic and geometric information. For the same task, SGPR [11] builds a graph representation of point clouds, which are enriched by both semantic and instance segmentation and perform graph similarity matching. SA-LOAM [2] integrates a semantic-aided variant of ICP into the popular LOAM pipeline for point cloud registration. To address loop closure, it uses a similar graph

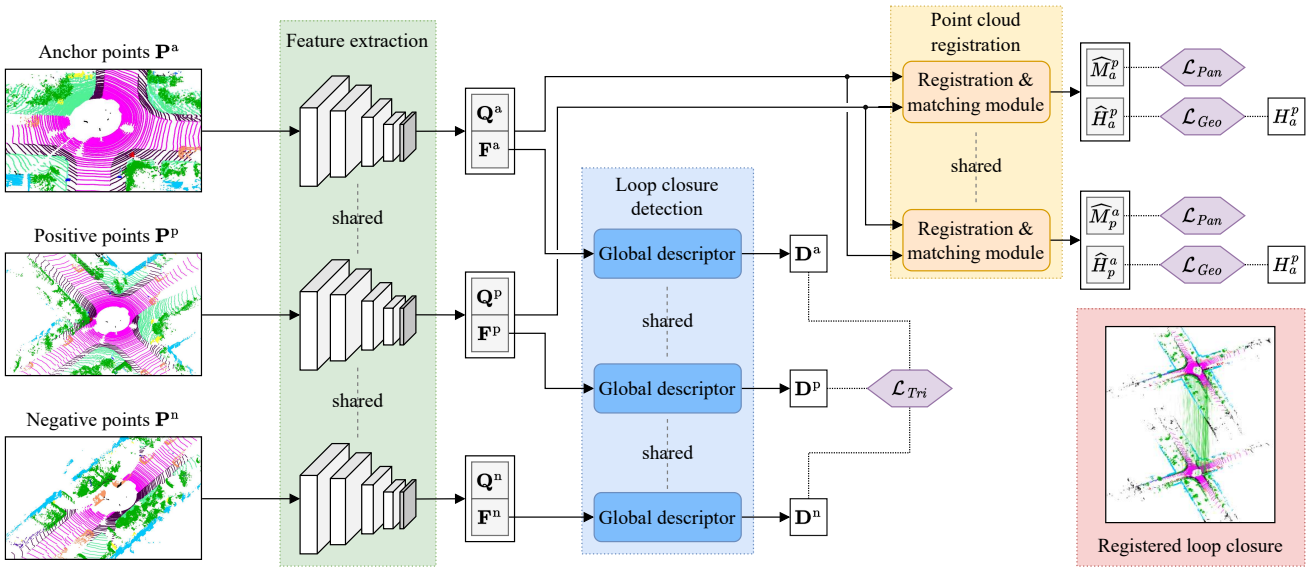


Fig. 2. Overview of our proposed PADLoC architecture for joint loop closure detection and point cloud registration. It consists of a shared feature extractor (green) followed by a global descriptor head (blue) for loop closure detection and a registration and matching module (orange) to estimate the 6-DoF transform between two point clouds (red). To train the global descriptor, we use a triplet loss (purple) that compares the anchor point cloud with a positive and negative sample. For training the registration module, we leverage losses (purple) based on both geometric and panoptic information. Note that during inference, no panoptic annotations are required, making PADLoC more versatile than other methods.

representation as Kong *et al.* [11]. SV-Loop [13] is a loop closure detection method for vision-based SLAM. It separately proposes loop closure candidates based on raw images and panoptic segmentation maps, which are then fused to extract the most feasible candidates. In our approach, we exploit panoptic annotations of point clouds while predicting both loop closure detection and point cloud registration. Additionally, we only utilize them during the training process but not for deployment, making the method more versatile.

III. Technical Approach

In this section, we introduce our novel PADLoC architecture for joint loop closure detection and point cloud registration. First, we detail the overall approach comprising the modules shown in Fig. 2. We then describe the loss functions that we employ, including our proposed loss that leverages panoptic annotations of point clouds.

A. Model Architecture

In this section, we describe the individual components of the PADLoC architecture. We build upon our previously proposed LCDNet [5], where instead of using a differentiable approximation of the optimal transport to obtain point matches, we propose to leverage the cross-attention matrices of transformers. The learnable keys, queries, and values weights yield a better latent representation of the features, and thus more reliable matches. As depicted in Fig. 2, the overall PADLoC architecture consists of three modules: feature extraction, loop closure detection, and point cloud registration. During training, we employ a triplet-based training scheme by feeding in an anchor point cloud along with a positive sample of a loop closure and a negative sample. Unlike other methods and as shown in Fig. 3, PADLoC does not require panoptic annotations during inference.

Feature Extraction: The feature extraction backbone converts raw input scans into a high-dimensional embedding that is

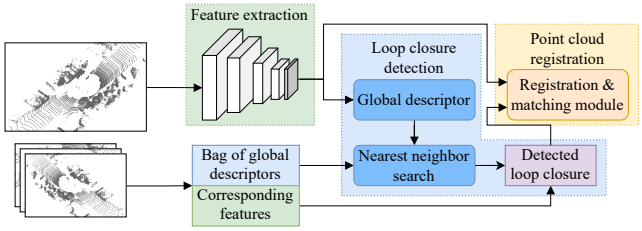


Fig. 3. During inference, PADLoC does not require panoptic annotations to extract features. To detect a loop closure, we perform a nearest neighbor search in the global descriptor space. If a loop is found, the 3D transformation is computed using the registration and matching module.

used as a common input for both loop closure detection and point cloud registration. It effectively exploits global and local contexts and is built upon the PV-RCNN architecture [32]. In detail, a point cloud $\mathbf{P}$, comprising 3D coordinates and reflectance values, is discretized into a voxel grid which is then passed through four sparse 3D convolutional layers to generate the feature maps at different resolutions. The final feature map is then stacked to form a BEV feature map. Additionally, the original point cloud is downsampled using the Farthest Point Sampling (FPS) algorithm to uniformly select n keypoints. The feature vector of each sampled keypoint is assembled by combining the feature maps from each convolutional layer in a neighborhood of the sampled keypoint using the Voxel Set Abstraction module [32]. The raw input of each sampled keypoint is also appended to each feature vector, along with the corresponding entry in the BEV feature map. Finally, these intermediate features are fed through a multilayer perceptron to obtain the final feature vector for each sampled point. This module thus outputs the sampled keypoints $\mathbf{Q}$ and the corresponding features $\mathbf{F}$.

Loop Closure Detection: The global descriptor module of PADLoC further encodes the previously extracted features to perform loop closure detection. For this task, we employ the NetVLAD layer [33] to convert the feature vectors $\mathbf{F}$ of the

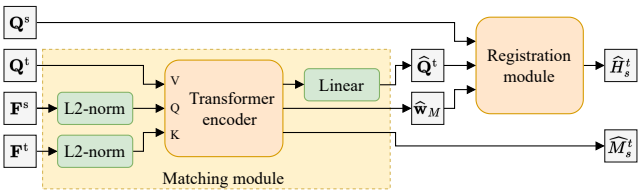


Fig. 4. The matching module consists of a transformer encoder that takes the extracted features of the source keypoints $\mathbf{F}^s$ as query, the features of the target keypoints $\mathbf{F}^t$ as key, and the corresponding target keypoints $\mathbf{Q}^t$ as value. It outputs both soft correspondences $\widehat{M}_s^t$ and projected target points $\widehat{\mathbf{Q}}^t$ along with confidence weights $\widehat{w}_M$. The latter is fed together with the source keypoints $\mathbf{Q}^s$ to a registration module that performs weighted SVD to estimate the final transform $\widehat{H}_s^t$.

anchor, the positive, and the negative points to their respective final descriptor $\mathbf{D}$. In detail, NetVLAD learns k clusters along with corresponding descriptors, which are aggregated in a single descriptor v for the entire point cloud. The final descriptors $\mathbf{D}$ of length g are then obtained via a context gating layer. This learnable pooling operation with weights $\mathbf{W}_G$ and bias $\mathbf{b}_G$ is defined as

$$\mathbf{D} = \sigma\big(\mathbf{W}_G \cdot v + \mathbf{b}_G\big) \odot v, \tag{1}$$

where $\sigma(\cdot)$ refers to the logistic sigmoid function and $\odot$ denotes the element-wise multiplication.

During inference, the descriptors are stored in such a manner that allows for efficient querying of the nearest neighbor in descriptor space. If the distance between the descriptor of the current scan and its nearest neighbor is below a predefined threshold, they are considered to form a loop closure. To avoid matching consecutive scans, we introduce a small temporal distance between the current scan and potential neighbors.

Point Matching: The matching module shown in Fig. 4 predicts soft correspondences $\widehat{M}_s^t$ between keypoints $\mathbf{Q}^s$ and $\mathbf{Q}^t$ of a source point cloud s and a target point cloud t, respectively. Additionally, it outputs projected target coordinates $\widehat{\mathbf{Q}}^t$ which are linear combinations of the original target coordinates with a one-to-one pairing with the points of the source set and a confidence weight $\widehat{w}_M$ for each of these matches. Inspired by the success of transformers in related tasks, we propose a novel architecture that performs cross-attention directly on the encoder part, obviating the need for a decoder by feeding independent inputs for the queries, keys, and values.

$$\widehat{\mathbf{Q}}^t = \mathbf{W}_Q \cdot \mathrm{TEL}\left(\mathbf{F}^s, \mathbf{F}^t, \mathbf{Q}^t\right) + \mathbf{b}_Q, \tag{2}$$

where $\mathrm{TEL}(q, k, v)$ is a transformer encoder layer, as defined in [34], but applied to independent query q, key k, and value v inputs. $\mathbf{W}_Q \in \mathbb{R}^{3\times f}$ and $\mathbf{b}_Q \in \mathbb{R}^3$ are learnable weights and biases used to reduce the dimensionality of the output from the size f of the features $\mathbf{F}$ to 3D space. We directly use the encoder's attention matrix as our matching $\widehat{M}_s^t$, since it already encodes the similarity between the features of the two sets of points. The output of the transformer encoder is given by the matrix product of the attention matrix and the weighted values input. Since in our case, we supply the target coordinates as the value input, it follows that the output of the transformer encoder corresponds to linear combinations of the input target points, weighted by the attention matrix and denoted by $\widehat{\mathbf{Q}}^t$. These projected points have a one-to-one correspondence with those of the anchor point cloud. Moreover, each row in the attention matrix represents the probability distribution of matching the corresponding point from the source set to all of the points from the target set, given that it is non-negative and adds up to one due to the use of the softmax function.

From the matching matrix $\widehat{M}_s^t$, we compute a confidence weight for every pair of point correspondences by penalizing the dispersion of the distributions represented by each row. We propose using a diversity metric for that purpose, such as the Shannon Entropy (E), the order-r Hill number (D^r), or the Berger-Parker index (BP), defined as

$$\mathrm{E}(\mathbf{p}) = -\sum_i p_i \cdot \log(p_i), \tag{3}$$

$$\mathrm{D}^r(\mathbf{p}) = \Big(\sum_i p_i^r\Big)^{\frac{1}{1-r}}, \tag{4}$$

$$\mathrm{BP}(\mathbf{p}) = \max(\mathbf{p}), \tag{5}$$

where $\mathbf{p}$ is a vector of probabilities.

The weights $\widehat{w}_M$ are obtained using either of the aforementioned metrics by normalizing their output to a $[0, 1]$ range, where the two extreme weights of 0 and 1 respectively correspond to a uniform and an infinitely sharp distribution.

Point Cloud Registration: To obtain the final relative transformation $\widehat{H}_s^t$ from a source point cloud to a target point cloud, we perform a weighted version of the Kabsch-Umeyama algorithm that finds the optimal translation and rotation between two sets of points by minimizing the root mean square error of the point pairs. First, the correspondences between the sampled source keypoints Q^s and the projected target keypoints $\widehat{Q}^t$ are weighted by the matching confidences $\widehat{w}_M$. Subsequently, the optimal translation is computed as the difference between the weighted centroids of the two point clouds. Finally, the optimal rotation is obtained via SVD of the weighted covariance matrix of the two sets of keypoints. This approach is fully differentiable and thus allows end-to-end training by measuring the error of the predicted transformation with respect to the ground truth relative pose.

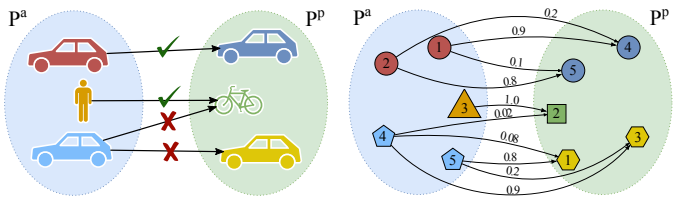
B. Loss Functions

Our total loss function consists of a weighted sum of the triplet loss $\mathcal{L}_{Tri}$ for loop closure detection as well as a geometric loss $\mathcal{L}_{Geo}$ and the newly proposed panoptic loss $\mathcal{L}_{Pan}$ for point cloud registration. The following paragraphs describe these losses in greater detail.

Triplet Loss: For the loop closure detection task, we use the triplet loss. It enforces a small distance between the descriptors of an anchor point cloud and a positive point cloud, i.e., a loop closure LiDAR scan while increasing the distance between the descriptors of the anchor and a negative point cloud, i.e., a LiDAR scan taken at a different place.

$$\mathcal{L}_{Tri} = \max\left\{d(\mathbf{D}^{\mathrm{a}}, \mathbf{D}^{\mathrm{p}}) - d(\mathbf{D}^{\mathrm{a}}, \mathbf{D}^{\mathrm{n}}) + m, 0\right\}, \tag{6}$$

where the descriptors of the anchor, the positive, and the negative sample are denoted by $\mathbf{D}^{\mathrm{a}}$, $\mathbf{D}^{\mathrm{p}}$, and $\mathbf{D}^{\mathrm{n}}$, respectively.



(a) Object matching (b) Graph representation

Fig. 5. The multi-matched object loss penalizes matching an object in the anchor point cloud to multiple objects in the positive sample. Unlike the semantic misclassification losses, the multi-matched object loss does not consider the semantic class, as depicted in (a). By exploiting a graph representation shown in (b) of the point cloud, it enforces that all points of the same object are matched to points of another object.

$d(\cdot)$ is a given distance function and m refers to the desired separation margin.

Geometric Loss: We formulate our geometric loss $\mathcal{L}_{Geo}$ as a sum of a pose loss $\mathcal{L}_{Pos}$ and an auxiliary matching loss $\mathcal{L}_{Mat}$. For the pose loss, we compare the predicted relative transformation $\widehat{H}_a^p$ from the anchor to the positive sample with the ground truth transformation H_a^p by applying both to the coordinates of the same sampled point cloud $\mathbf{Q}^{\mathrm{a}}$. Then we compute the mean absolute error in the Euclidean space.

$$\mathcal{L}_{Pos} = \text{mean}\left(\text{abs}\left(\widehat{H}_a^p \cdot \mathbf{Q}^{\mathrm{a}} - H_a^p \cdot \mathbf{Q}^{\mathrm{a}}\right)\right) \tag{7}$$

We further evaluate the geometric correspondence between the sampled anchor $\mathbf{Q}^{\mathrm{a}}$ and positive points $\mathbf{Q}^{\mathrm{p}}$ leveraging the predicted matching matrix $\widehat{M}_p^a$. In detail, we transform the anchor points with the ground truth transformation H_p^a and project the positive sample with $\widehat{M}_p^a$.

$$\mathcal{L}_{Mat} = \text{mean}\left(\text{abs}\left(H_a^p \cdot \mathbf{Q}^{\mathrm{a}} - \widehat{M}_p^a \cdot \mathbf{Q}^{\mathrm{p}}\right)\right) \tag{8}$$

Panoptic Loss: In addition to the geometric point correspondences, we propose to leverage panoptic information to register two point clouds. In detail, we formulate a novel panoptic loss $\mathcal{L}_{Pan}$ as the sum of semantic misclassification losses $\mathcal{L}_{Sem}$ and $\mathcal{L}_{Mes}$ as well as a multi-matched object loss $\mathcal{L}_{Mmo}$.

We treat the matching process as a classification task, where the projected positive points are assigned a semantic class. While a cross-entropy loss is commonly used in classification problems, due to the fact that the proposed class logits are not the output of either a logistic or softmax activation, we empirically found that a mean absolute error resulted in a more stable training process. First, we use the semantic labels to construct one-hot encoded matrices $\mathbf{K}^{\mathrm{a}}$ and $\mathbf{K}^{\mathrm{p}}$ for the anchor and positive samples, respectively. Using the predicted matching matrix $\widehat{M}_p^a$, we define the semantic loss as

$$\mathcal{L}_{Sem} = \text{mean}\left(\text{abs}\left(\mathbf{K}^{\mathrm{a}} - \widehat{M}_p^a \cdot \mathbf{K}^{\mathrm{p}}\right)\right). \tag{9}$$

Additionally, to allow flexibility in the semantic misclassification, we define a mapping from the semantic class labels to a set of super-classes, e.g., both *car* and *truck* belong to the *vehicle* class. Further details can be found in Sec. IV-A. Analogously to the semantic loss, we construct one-hot encoded matrices $\mathbf{J}^a$ and $\mathbf{J}^p$ and define the meta-semantic loss as

$$\mathcal{L}_{Mes} = \text{mean}\left(\text{abs}\left(\mathbf{J}^{\mathrm{a}} - \widehat{M}_p^a \cdot \mathbf{J}^{\mathrm{p}}\right)\right). \tag{10}$$

In our novel multi-matched object loss, we further exploit the instance labels to encourage the network to match entire objects consistently from one point cloud to the other. This is done by penalizing matches of points from a single object in the anchor to multiple objects in the positive sample. Unlike the previously introduced semantic misclassification losses, the multi-matched object loss does not consider the semantic class of objects, as depicted in Fig. 5 (a).

Since instance labels may not be consistent throughout a driving sequence, it is not feasible to purely rely on the IDs. Therefore, we construct adjacency matrices $\mathbf{O}^{\mathrm{a}}$ and $\mathbf{O}^{\mathrm{p}}$ of a graph representation of the point clouds, where nodes represent points and edges connect points of the same instances of a semantic class. The predicted matching matrices $\widehat{M}_p^a$ and $\widehat{M}_a^p$ can then be considered as weighted, directed, bipartite graphs between the two sets of points (see Fig. 5 (b)). Finally, we formulate the multi-matched object loss as

$$\mathcal{L}_{Mmo} = \text{mean}\left((1 - \mathbf{O}^{\mathrm{a}}) \odot \left(\widehat{M}_p^a \cdot \mathbf{O}^{\mathrm{p}} \cdot \widehat{M}_a^p\right)\right), \tag{11}$$

where $\odot$ denotes the element-wise multiplication.

Reverse Losses: Finally, we add a second instance of the registration module that processes the swapped source s and target t inputs and predicts the inverse relative transformation. Both the geometric and the panoptic losses can be reformulated accordingly. The total loss is then formulated by averaging the results of both the original and the reverse versions.

IV. Experimental Evaluation

In this section, we evaluate our proposed PADLoC architecture with respect to multiple handcrafted and learning-based methods. We perform several experiments and present both the loop closure detection and the point cloud registration results. Finally, we evaluate the design choices in PADLoC by performing multiple ablation studies and provide a brief efficiency analysis.

A. Implementation Details

We perform experiments on two publicly available autonomous driving datasets, namely the KITTI odometry benchmark [14] and the Ford campus vision and LiDAR dataset [15]. Additionally, we also present results on a more challenging in-house dataset recorded in Freiburg, Germany. For training PADLoC, we leverage the ground truth panoptic annotations from the SemanticKITTI dataset [38]. If not specified otherwise, we train all learning-based models on sequences {00, 05, 06, 07, 09} of KITTI and evaluate on sequence 08. For the results on the Ford and Freiburg datasets presented in Table I, we do not retrain the methods but use the weights trained on KITTI. Unless otherwise specified, we use $n = 4096$ keypoints, set the feature size to $f = 640$, the descriptor length to $g = 256$, and the number of clusters $k = 64$. To improve the invariance of the model with respect to the inputs' position and orientation, we augment the data during training

TABLE I
COMPARISON OF LOOP CLOSURE DETECTION AND POINT CLOUD REGISTRATION PERFORMANCE

		KITTI Seq. 08 [14]					Ford Seq. 01 [15]					Freiburg *(in-house)*				
	Method	AP	Max-F1	EP	r_{err} [°]	t_{err} [m]	AP	Max-F1	EP	r_{err} [°]	t_{err} [m]	AP	Max-F1	EP	r_{err} [°]	t_{err} [m]
Handcrafted	M2DP [19]	0.05	0.10	0.00	—	—	0.89	0.88	0.89	—	—	0.71	0.68	0.74	—	—
	Scan Context* [35]	0.65	0.62	0.00	3.11	—	0.97	**0.95**	0.94	16.68	—	0.81	**0.79**	**0.82**	52.70	—
	LiDAR-Iris* [36]	0.64	0.62	**0.71**	1.84	—	0.90	0.64	0.50	1.66	—	0.81	0.78	**0.82**	46.24	—
	ISC* [21]	0.31	0.32	0.55	6.27	—	0.62	0.70	0.00	6.15	—	0.82	0.75	0.79	51.02	—
	ICP (pt2pt) [9]	—	—	—	160.63	2.41	—	—	—	9.56	2.79	—	—	—	89.43	2.37
	ICP (pt2pl) [9]	—	—	—	160.73	2.49	—	—	—	9.16	2.62	—	—	—	89.25	2.25
Learning	DCP [6]	—	—	—	46.06	2.59	—	—	—	12.14	3.42	—	—	—	45.70	2.30
	SGPR [11]	0.06	0.13	0.00	—	—	0.11	0.27	0.01	—	—	0.15	0.31	0.05	—	—
	OverlapNet* [25]	0.32	0.37	0.50	65.45	—	0.79	0.81	0.84	9.44	—	0.76	0.72	0.76	70.91	—
	LCDNet [5]	0.76	0.74	0.50	**0.37**	0.19	0.97	0.93	0.72	1.82	1.44	**0.84**	0.73	0.71	10.08	**0.91**
	PADLoC (ours)	**0.81**	**0.78**	0.51	**0.37**	**0.16**	**0.98**	0.85	**0.95**	**1.50**	**1.33**	0.83	0.74	0.74	**9.30**	1.41

Comparison of the average precision (AP), the maximum F1 score, and the extended precision (EP) [37] for loop closure detection as well as rotation error r_{err} and translation error t_{err} for point cloud registration of PADLoC with previous methods. All learning-based models are trained on the KITTI odometry benchmark dataset. PADLoC uses panoptic annotations from the SemanticKITTI dataset. Methods denoted with * only estimate the yaw between two point clouds instead of a full 6-DoF transformation. Bold and underlined values denote the best and second best scores, respectively.

by applying a random rigid transformation to the input point clouds with a uniform translation of $\pm 1.5\,\mathrm{m}$ in the x and y axes and $\pm 0.25\,\mathrm{m}$ along z, and a uniform rotation of $\pm 3°$ for the roll and pitch angles and $\pm 180°$ for the yaw. We train all our models on a server with 4 NVIDIA RTX A6000 GPUs for 150 epochs with a batch size of $b = 8$. We use the Adam optimizer with an initial learning rate of $\lambda = 0.004$, halved after epochs 40 and 80, and with a weight decay of 5×10^{-6}.

The total loss function is computed as a weighted sum of the components described in Sec. III-B, with weights $w_{Tri} = 1.0$, $w_{Pos} = 1.0$, $w_{Mat} = 0.05$, $w_{Sem} = 0.125$, $w_{Mes} = 0.5$, and $w_{Mmo} = 10.0$. We use a triplet margin of $m = 0.5$ and the L2 distance as the distance function in Eq. 6. For the semantic super-classes, we follow the definitions of Cityscapes [39] and group the semantic labels into *flat*, *human*, *vehicle*, *construction*, *object*, *nature*, and *void*. Based on the ablation study presented in Sec. IV-D, we use the Berger-Parker index to compute the confidence weights.

B. Loop Closure Detection

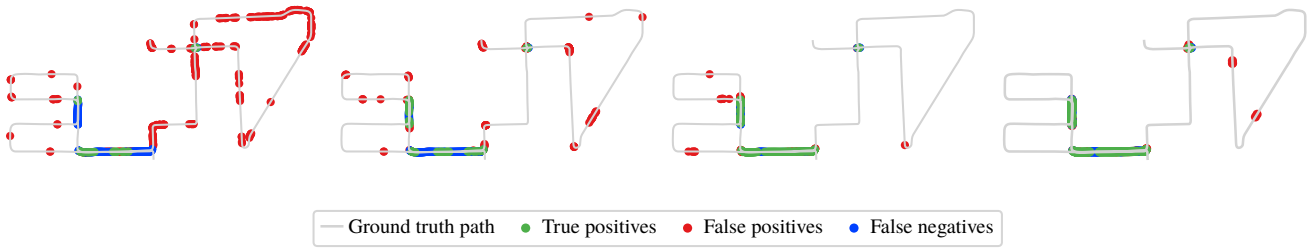
To evaluate the loop closure detection performance, we compare PADLoC with the handcrafted methods M2DP [19], Intensity Scan Context (ISC) [21], Scan Context [35], and LiDAR-Iris [36], as well as with the learning-based approaches LCDNet [5], OverlapNet [25], Deep Closest Point (DCP) [6], and SGPR [11], which uses panoptic information also during inference. For DCP, we combine the feature extraction module of PADLoC with a full transformer-based matching module based on the authors' code release. For the other methods, we directly use the official code published by the respective authors. To compute the results on OverlapNet and SGPR, we download the model weights provided on the project website that are trained on KITTI. Since SGPR requires panoptic labels during inference time, we use predictions by RangeNet++ [40] combined with point clustering to obtain instances.

When evaluating PADLoC, we generate a descriptor $\mathbf{D}_i$ for every scan i in a sequence and compute its similarity with that of all frames prior to the 50 previous scans. If a scan j with the closest descriptor to that of scan i has a similarity higher than a threshold τ, then the pair (i, j) is considered to form a loop closure. If the distance between the two ground truth poses is within $4\,\mathrm{m}/10\,\mathrm{m}/20\,\mathrm{m}$ for the KITTI/Ford/Freiburg dataset, then it is considered a true positive. Otherwise, it is considered a false positive. Conversely, if the pose distance is within $4\,\mathrm{m}/10\,\mathrm{m}/20\,\mathrm{m}$, but the similarity between the descriptors is below the threshold τ, then we regard it as a false negative. By changing the value of τ, we obtain precision-recall pairs that are then used to compute the average precision (AP).

In Table I, we report the AP, the maximum F1 score, and the extended precision (EP) [37] of PADLoC and the aforementioned baseline methods. Notably, PADLoC achieves the highest AP and Max-F1 score across the entire board for the evaluation sequences of KITTI and the highest AP and EP on Ford. On our in-house Freiburg dataset, PADLoC yields the highest Max-F1 score as well as the second best AP and EP compared to the other learning-based approaches. Although the proposed transformer-based registration head and the panoptic losses do not directly influence the loop closure detection module, by sharing the same feature extractor between the two branches and jointly training the two tasks, PADLoC learns a better feature representation improving the loop closure detection performance compared to LCDNet, which achieved the second best AP on both KITTI and Ford. Qualitative results of these methods on the KITTI dataset are visualized in Fig. 6. Compared to OverlapNet, both LCDNet and PADLoC correctly detect a higher number of loop closures, whereas PADLoC is able to further reduce the number of false positives. While the learning-based methods LCDNet and PADLoC outperform all handcrafted methods when evaluated on the same domain as used for training, this gap vanishes on Ford and Freiburg. Here, these methods perform on par with the best handcrafted approach Scan Context.

C. Point Cloud Registration

To evaluate the point cloud registration performance, we compare PADLoC with the same handcrafted and learning-based methods described in Sec. IV-B, except for M2DP and SGPR that do not perform point cloud registration. Since the handcrafted methods only estimate the yaw between two point clouds instead of the full 6-DoF transformation, we additionally compare with the Iterative Closest Point algorithm (ICP) [9], using both point-to-point and point-to-plane



(a) SGPR [11] (b) OverlapNet [25] (c) LCDNet [5] (d) PADLoC (ours)

Fig. 6. Qualitative loop closure detection results on KITTI sequence 08 of the learning-based methods. The ground truth path corresponds to true negatives. While LCDNet reduces both false positives and false negatives compared to OverlapNet, the proposed PADLoC further decreases false positives.

TABLE II
ABLATION STUDY ON CONFIDENCE WEIGHTS

Method	AP ↑	r_{err} [°] ↓	t_{err} [m] ↓
Uniform	0.73	4.63	3.76
Column sum	0.76	6.34	3.62
Shannon	0.50	21.86	3.99
Hill (r=2)	**0.89**	2.45	2.00
Hill (r=4)	0.84	2.47	2.12
Berger-Parker	0.81	**2.35**	**1.43**

Average precision (AP) of loop closure detection as well as the mean error of point cloud registration, evaluated on KITTI sequence 08 for different weightings used in SVD.

TABLE III
INFLUENCE OF THE LOSS FUNCTIONS

$\mathcal{L}_{Geo}$	$\mathcal{L}_{Pan}$	$\mathcal{L}_{Rev}$	AP ↑	r_{err} [°] ↓	t_{err} [m] ↓
✓			0.70	3.09	1.62
✓	✓		0.78	3.36	1.71
✓	✓	✓	**0.81**	**2.35**	**1.43**

Average precision (AP) of loop closure detection and the mean error of point cloud registration, evaluated on KITTI sequence 08 for the different loss functions.

distances. Following the standard experimental setup [5], for LCDNet, DCP, and PADLoC, we perform point cloud registration with RANSAC using the extracted features before the respective matching layers.

As a measure of registration accuracy, we compute the rotation error r_{err} in degrees and the translation error t_{err} in meters of all positive pairs. We then average the errors over the entire sequence and present the results in Table I. We observe that PADLoC yields the smallest rotation error compared to all the handcrafted and learning-based methods on each of the evaluation sequences in the datasets. Additionally, it yields the smallest translation error on both the KITTI and Ford datasets, as well as the second lowest translation error on our in-house Freiburg dataset. LCDNet achieves the second best performance in most evaluations while achieving the lowest translation error on the Freiburg dataset. This result shows that while the feature extraction architecture and the training scheme play an important role, leveraging the cross-modal attention matrices from the transformer architecture and the panoptic information during training further improves the point cloud registration performance. While LiDAR-Iris achieves the lowest rotation error across all the handcrafted methods, it only estimates the yaw angle instead of the full 6-DoF transformation.

D. Ablation Studies

In this section, we present ablation studies to analyze the major design choices of PADLoC. As the RANSAC-based point cloud registration described in Sec. IV-C is applied only during inference and does not impact the training stage, the experiments in this section do not exploit RANSAC.

Confidence Weighting: We investigate the effect of different weighting schemes on the performance of both loop closure detection and point cloud registration tasks. In Table II, we present the average precision (AP) as well as the registration errors r_{err} and t_{err} for the six weighting methods. In particular, uniform weights corresponding to unweighted SVD, column sum representing the method used in LCDNet [5], where weights are the sums along the columns of the matching matrix, and the diversity metrics from Sec. III-A, i.e., the Shannon Entropy, the order-r Hill number with $r \in \{2, 4\}$, and the Berger-Parker index. We observe that both the Hill numbers and the Berger-Parker index outperform the other confidence weighting methods. Due to the substantially smaller translation error of the Berger-Parker index, improving the registration by more than $0.5\,\mathrm{m}$, we use this method in our final design.

Effect of Losses: To demonstrate the efficacy of our proposed panoptic loss $\mathcal{L}_{Pan}$ and the impact of formulating all losses in a bidirectional manner ($\mathcal{L}_{Rev}$), we consecutively add them to the original geometric loss $\mathcal{L}_{Geo}$. We present the results for both the loop closure detection and point cloud registration tasks in Table III. We observe that adding the proposed panoptic losses increases the average loop closure detection precision by further constraining which points can be matched together based on their semantic and instance labels. Furthermore, by including the second matching and registration head, along with its corresponding reverse losses as illustrated in the bottom row, the added bidirectional consistency constraint yields the highest AP and the smallest registration errors.

E. Efficiency Analysis

We evaluate the memory footprint and inference time of our method on an NVIDIA RTX 3090 GPU. PADLoC requires $3.4\,\mathrm{GB}$. Compared to the full transformer-based matching module of DCP that requires $10.3\,\mathrm{GB}$, the memory footprint of PADLoC is only one-third, showing its lower complexity.

On average, PADLoC needs $10\,\mathrm{ms}$ for pre-processing a single point cloud. The shared feature extraction step consumes $167\,\mathrm{ms}$. Computing the global descriptor used for loop closure detection takes $0.1\,\mathrm{ms}$ per point cloud. Finally, one forward pass of the registration and matching module to compute the transform between two point clouds takes $14\,\mathrm{ms}$.

V. CONCLUSION

In this paper, we proposed the novel PADLoC architecture for LiDAR-based joint loop closure detection and point cloud

registration. PADLoC is composed of a common feature extractor, a global descriptor as well as a transformer-based registration and matching module. Unlike previous approaches, we feed different inputs as value, query, and key to the transformer encoder exploiting its internal structure. We further introduced a new loss function that leverages ground truth panoptic annotations by penalizing matching points from different semantic classes as well as across multiple objects and validated its positive impact. Through extensive experimental evaluations, we demonstrated the efficacy of PADLoC compared to both handcrafted and learning-based methods. Future work will focus on exploiting panoptic information in an online manner and applying the matching approach of PADLoC to point cloud registration tasks in other domains.

REFERENCES

[1] N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Continual SLAM: Beyond lifelong simultaneous localization and mapping through continual learning," in *Int. Symposium of Robotics Research*, 2022.

[2] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "SA-LOAM: Semantic-aided LiDAR SLAM with loop closure," in *Int. Conf. on Robotics and Automation*, 2021, pp. 7627–7634.

[3] B. Bešić and A. Valada, "Dynamic object removal and spatio-temporal RGB-D inpainting via geometry-aware adversarial learning," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 170–185, 2022.

[4] N. Gosala and A. Valada, "Bird's-eye-view panoptic segmentation using monocular frontal view images," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1968–1975, 2022.

[5] D. Cattaneo, M. Vaghi, and A. Valada, "LCDNet: Deep loop closure detection and point cloud registration for LiDAR SLAM," *IEEE Transactions on Robotics*, pp. 1–20, 2022.

[6] Y. Wang and J. Solomon, "Deep Closest Point: Learning representations for point cloud registration," in *Int. Conf. on Computer Vision*, 2019, pp. 3522–3531.

[7] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022, pp. 11 143–11 152.

[8] Z. J. Yew and G. H. Lee, "REGTR: End-to-end point cloud correspondences with transformers," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, June 2022, pp. 6677–6686.

[9] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[10] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 4530–4537.

[11] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3D point clouds," in *Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 8216–8223.

[12] N. Radwan, A. Valada, and W. Burgard, "VLocNet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, 2018.

[13] Z. Yuan, K. Xu, B. Deng, X. Zhou, P. Chen, and Y. Ma, "SV-Loop: Semantic-visual loop closure detection with panoptic segmentation," in *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, 2021, pp. 245–250.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[15] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Int. Conf. on Robotics and Automation*, 2009, pp. 3212–3217.

[17] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3D range scans taking into account object boundaries," in *Int. Conf. on Robotics and Automation*, 2011, pp. 2601–2608.

[18] L. Luo, S.-Y. Cao, Z. Sheng, and H.-L. Shen, "LiDAR-based global localization using histogram of orientations of principal normals," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022.

[19] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 231–237.

[20] G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 4802–4809.

[21] H. Wang, C. Wang, and L. Xie, "Intensity Scan Context: Coding intensity and geometry relations for loop closure detection," in *Int. Conf. on Robotics and Automation*, 2020, pp. 2095–2101.

[22] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "SSC: Semantic scan context for large-scale place recognition," in *Int. Conf. on Intelligent Robots and Systems*, 2021, pp. 2092–2099.

[23] M. Angelina Uy and G. Hee Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[25] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "OverlapNet: A siamese network for computing lidar scan similarity with applications to loop closing and localization," *Autonomous Robots*, 2021.

[26] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6958–6965, 2022.

[27] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer graphics forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using PointNet," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[30] A. L. Ballardini, D. Cattaneo, and D. G. Sorrenti, "Visual localization at intersections with digital maps," in *Int. Conf. on Robotics and Automation*, 2019, pp. 6651–6657.

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

[32] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 10 526–10 535.

[33] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, vol. 30, 2017, pp. 6000–6010.

[35] G. Kim, S. Choi, and A. Kim, "Scan Context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, 2022.

[36] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "LiDAR Iris for loop-closure detection," in *Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 5769–5775.

[37] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1688–1695, 2020.

[38] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Int. Conf. on Computer Vision*, 2019.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[40] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 4213–4220.