

LCDNet: Deep Loop Closure Detection and Point Cloud Registration for LiDAR SLAM

Daniele Cattaneo¹, Matteo Vaghi², Abhinav Valada¹

Abstract—Loop closure detection is an essential component of Simultaneous Localization and Mapping (SLAM) systems, which reduces the drift accumulated over time. Over the years, several deep learning approaches have been proposed to address this task, however their performance has been subpar compared to handcrafted techniques, especially while dealing with reverse loops. In this paper, we introduce the novel LCDNet that effectively detects loop closures in LiDAR point clouds by simultaneously identifying previously visited places and estimating the 6-DoF relative transformation between the current scan and the map. LCDNet is composed of a shared encoder, a place recognition head that extracts global descriptors, and a relative pose head that estimates the transformation between two point clouds. We introduce a novel relative pose head based on the unbalanced optimal transport theory that we implement in a differentiable manner to allow for end-to-end training. Extensive evaluations of LCDNet on multiple real-world autonomous driving datasets show that our approach outperforms state-of-the-art loop closure detection and point cloud registration techniques by a large margin, especially while dealing with reverse loops. Moreover, we integrate our proposed loop closure detection approach into a LiDAR SLAM library to provide a complete mapping system and demonstrate the generalization ability using different sensor setup in an unseen city.

Index Terms—Loop Closure Detection, Point Cloud Registration, Place Recognition, Simultaneous Localization and Mapping, Deep Learning.

I. INTRODUCTION

SIMULTANEOUS Localization and Mapping (SLAM) is an essential task for autonomous mobile robots as it is a critical precursor for other tasks in the navigation pipeline. A failure in the SLAM system will adversely affect all the subsequent tasks and negatively impact the functioning of the robot. Therefore, improving the robustness of SLAM systems has garnered significant interest from both industry and academia in the past decades, as demonstrated by the widespread adoption in many fields such as self-driving cars [1], unmanned aerial vehicles [2], agricultural robots [3], and autonomous marine vehicles [4]. The goal of any SLAM system is to build a map of an unknown environment by exploiting onboard sensor data (such as Global Positioning System (GPS), cameras, Light Detection and Ranging (LiDAR), and Inertial Measurement Units (IMUs)), and at the same time localize the robot within the built map.

A typical SLAM pipeline consists of three main components: (i) consecutive scan alignment in which subsequent scans are aligned by leveraging information such as odometry, scan

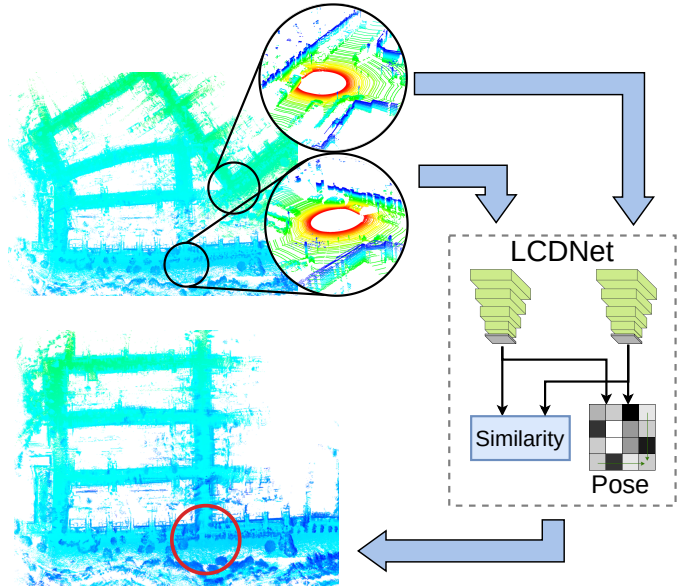


Figure 1. Our proposed LCDNet detects loops by computing the similarity between two point clouds and predicting the relative pose between them. This is a crucial component of any SLAM system, as it reduces the drift accumulated over time.

matching, or IMUs information; (ii) loop detection to identify places that were previously visited, and (iii) loop closure to align the current scan to the previously visited place and accordingly correct the map. Since the first step by itself usually drifts over time due to its incremental nature, the map will no longer be consistent when the robot navigates through a place that was previously visited. Therefore, steps (ii) and (iii) are employed to reduce the accumulated drift by adding a new constraint to the pose graph when a loop is detected. Subsequently, all the previous poses are corrected according to this constraint, thus generating a consistent map.

Although several vision-based SLAM systems have been proposed [5], [6], their loop closure methods often fail in case of strong variation due to illumination, appearance, or viewpoint changes. LiDARs on the other hand, are invariant to illumination changes and provide an accurate geometric reconstruction of the surrounding environment. Hence, they are often preferred over cameras for SLAM approaches due to their inherent robustness. Standard LiDAR-based loop detection methods extract local keypoints [7], [8] or global handcrafted descriptors [9], [10], and compare the descriptor of the current scan with that of previous scans to identify loops. However, most of these approaches require an ad hoc function to compare

¹ Department of Computer Science, University of Freiburg, Germany.

² Department Informatica, Sistemistica e Comunicazioni, Università degli studi di Milano - Bicocca, Italy.

the descriptors of two point clouds, which drastically impacts the runtime when the number of past scans increases.

Driven by the significant strides achieved by Deep Neural Networks (DNNs) in different fields, many recent works [11], [12], [13], [14], [15] employ DNNs to address the loop detection task in LiDAR-based SLAM systems. While these approaches are generally faster than handcrafted methods, their performance is not on par with these state-of-the-art methods, especially in the case of reverse loops. Another challenge faced while detecting a loop closure is related to aligning the current point cloud with the built map. A common approach is to leverage standard techniques for scan matching such as the Iterative Closest Point (ICP) [16] algorithm or one of its variants [17], [18], [19]. Although ICP is generally able to successfully align two point clouds when they are relatively similar and close, it can fall in local minima when the initial pose between the point clouds is very different. This is often the case when faced with reverse direction loops. To overcome this problem, some methods also provide an estimate of the rotation between the two point clouds. This estimate can then be used as an initial guess in the ICP algorithm to aid the alignment to converge to the correct solution.

Several recent works [20], [21], [22], [23] have been proposed to address the point cloud registration task by leveraging the advancement in deep learning. Although these approaches achieve impressive results in registering single objects and outperform standard techniques, the protocol used to test these approaches only consider a relatively small initial rotation misalignment (up to 45°). However, the point clouds can be rotated by 180° in the loop closure task. Recent work has shown that some of these methods have a very low success rate when the initial misalignment is larger than 120° [20].

In this paper, we propose the novel LCDNet for loop closure detection which performs both loop detection and point cloud registration (see Figure 1). Our method combines the ability of DNNs to extract distinctive features from point clouds, with algorithms from the transport theory for feature matching. LCDNet is composed of a shared backbone that extracts point features, followed by the place recognition head that extracts global descriptors and the relative pose head that estimates the transformation between two point clouds. One of the core components of our LCDNet is the Unbalanced Optimal Transport (UOT) algorithm that we implement in a differentiable manner. UOT allows us to effectively match the features extracted from the two point clouds, reject outliers, and handle occluded points, while still being able to train the network in an end-to-end manner. As opposed to existing loop closure detection methods that estimate the relative yaw rotation between two point clouds, our proposed LCDNet estimates the full 6-DoF relative transformation under driving conditions between them which significantly helps the subsequent ICP refinement to converge faster.

We train our proposed LCDNet on sequences from the KITTI odometry [24] and KITTI-360 [25] datasets, and evaluate it on the unseen sequences on both datasets. Moreover, we found

that there is a lack of a standard protocol for evaluating loop closure detection methods in the existing literature. Different works evaluated their approaches using different metrics such as precision-recall curve, average precision, Receiver Operating Characteristic (ROC) curve, recall@k, and maximum F1-score. Even among the methods that use the same metric for evaluation, there are still substantial differences in the other parameters chosen for computing the metrics which makes the performance of existing methods not directly comparable. For example, the definition of a true loop can span from scans within three meters [14] up to scans within 15 meters [15]. Therefore, in this work, we evaluate existing state-of-the-art approaches using a uniform evaluation protocol to provide a fair comparison. Exhaustive comparisons demonstrate that our proposed LCDNet outperforms both handcrafted methods as well as DNN-based methods and achieves state-of-the-art performance on both loop closure detection and point cloud registration tasks. Furthermore, we present detailed ablation studies on the architectural topology of LCDNet and also present results from integrating LCDNet into a recent LiDAR SLAM library [26]. Additionally, we demonstrate the generalization ability of our proposed approach using experiments with a different sensor setup from an autonomous driving scenario in a completely different city.

The main contributions of this work can be summarized as follows:

- 1) We propose LCDNet, a novel approach for loop closure detection that effectively detects reverse loops.
- 2) We propose an end-to-end trainable relative pose regression network based on the unbalanced optimal transport theory that can register two point clouds that only partially overlap and with an arbitrary initial misalignment.
- 3) We comprehensively evaluate existing state-of-the-art loop closure detection methods using a uniform evaluation protocol, we perform extensive evaluations of LCDNet on multiple autonomous driving datasets, and we present detailed ablation studies that demonstrate the efficacy of our contributions.
- 4) We study the generalization ability of our approach to unseen environments and different sensor setups by evaluating LCDNet on our own recorded dataset around the city of Freiburg, Germany.
- 5) We integrate our network into a SLAM library to provide a complete system for localization and mapping and we make the code, the entire SLAM system, and the evaluation tools publicly available at <http://rl.uni-freiburg.de/research/lidar-slam-1c>.

The remainder of the paper is organized as follows: we review existing methods that are related to our approach in Section II. In Section III, we detail our proposed LCDNet and the integration into the SLAM system. We then present experiments that demonstrate the effectiveness and robustness of LCDNet in Section IV. Finally, we present our conclusions in Section V.

II. RELATED WORKS

In this section, we provide an overview of the state-of-the-art techniques for vision-based and LiDAR-based loop closure detection, followed by methods for point cloud registration.

Loop Closure Detection: Techniques for loop closure detection can primarily be categorized into visual and LiDAR-based methods. Traditionally, vision-based techniques for loop closure detection rely on handcrafted features for identifying and representing relevant parts of scenes depicted within images, and exploit a Bag-of-Words model to combine them [5], [6]. In the last few years, deep learning approaches [27], [28] have been proposed that achieve successful results. These techniques employ DNN for computing global descriptors to provide a compact representation of images and perform direct comparisons between descriptors for searching matches between similar places. Recently, [29] proposed a novel approach that employs DNNs for extracting local features from intermediate layers and organizes them in a word-pairs model. Although vision-based methods achieve impressive performance, they are not robust against adverse environmental situations such as challenging light conditions and appearance variations that can arise during long-term navigation. As loop closure detection is a critical task within SLAM systems, in this work, we exploit LiDARs for the sensing modality since they provide more reliable information even in challenging conditions in which visual systems fail.

3D LiDAR-based techniques have gained significant interest in the last decade, as LiDARs provide rich 3D information of the environment with high accuracy and their performance is not affected by illumination changes. Similar to vision-based approaches, LiDAR-based techniques also exploit local features. Most methods use 3D keypoints [30], [31] that are organized in a bag-of-words model for matching point clouds [7]. [8] propose a keypoint based approach in which a nearest neighbor voting paradigm is employed to determine if a set of keypoints represent a previously visited location. Recently, [32] propose a voxel-based method that divides a 3D scan into voxels and extracts multiple features from them through different modalities, followed by learning the importance of voxels and types of features.

Another category of techniques represents point clouds through global descriptors. [9] propose an approach that directly produces point clouds fingerprints. In particular, this method relies on density signatures extracted from multiple projections of 3D point clouds on different 2D planes. [10] introduces a novel global descriptor called Scan Context that exploits bird-eye-view representation of a point cloud together with a space partitioning procedure to encode the 2.5D information within an image. In a similar approach, [33] propose a method to extract binary signature images from 3D point clouds by employing LoG-Gabor filtering with thresholding operations to obtain a descriptor. The main drawback of these approaches is that they require an ad-hoc function to compare the global descriptor of two point clouds which drastically impacts the runtime when the number of scans to compare increases.

Recently, DNN-based techniques have also been proposed for computing descriptors from 3D point clouds. [11] propose PointNetVLAD which is composed of PointNet [34] with a NetVLAD layer [27] and yields compact descriptors. [12] propose OREOS which computes 2D projection of point clouds on cylindrical planes and is subsequently fed into a DNN that computes global descriptors and estimate their yaw discrepancy. More recently, the OverlapNet [13] architecture was introduced, which estimates the overlap and relative yaw angle between a pair of point clouds. The overlap estimate is then used for detecting loop closures while the yaw angle estimation is provided to the Iterative Closest Point (ICP) algorithm as the initial guess for the point clouds alignment. While DNN-based methods are generally faster than classical techniques, and show promising results in sequences that contain loops only in the same direction, their performance drastically decreases when they are faced with reverse loops.

Recently, techniques that exploit graph structures by matching semantic graphs have been proposed [14], [15]. These approaches first extract semantic information and perform instance retrieval, followed by defining graph vertices on the object centroids. Subsequently, features are extracted by considering handcrafted descriptors or by processing nodes through a Dynamic Graph CNN [35]. Finally, loop closures are identified by comparing vertices between graphs. However, computing the exact correspondences between two graphs is still an open problem and existing methods are only suitable when a few vertices are considered or they can only provide an approximated solution [36]. In this work, we exploit recent advancements in deep learning and propose a DNN-based approach for detecting loop closure by combining high-level voxel features with fine-grained point features. Our approach effectively detects loops in challenging scenarios such as reverse loops and outperforms state-of-the-art handcrafted and learning-based techniques.

Point Cloud Registration: Point clouds registration represents the task of finding a rigid transformation to accurately align a pair of point clouds. The ICP algorithm [16] is one standard method that is often employed to tackle this task. Although ICP is one of the most popular methods, the main drawback concern the initial rough alignment of point clouds which is required to reach an acceptable solution, and the algorithm complexity which increases drastically with the number of points. Other methods tackle the registration problem globally without requiring a rough initial alignment. Traditionally, these techniques exploit local features [37] for finding matches between point clouds and employ algorithms such as RANdom SAMple Consensus (RANSAC) [38] for estimating the final transformation. However, the presence of noise in the input data and outliers generated from incorrect matches can lead to an inaccurate result. To address these problems, [39] proposes a global registration approach that ensures fast and accurate alignment, even in the presence of many outliers.

Recent years have also seen the introduction of deep learning methods that tackle the registration problem. A typical approach

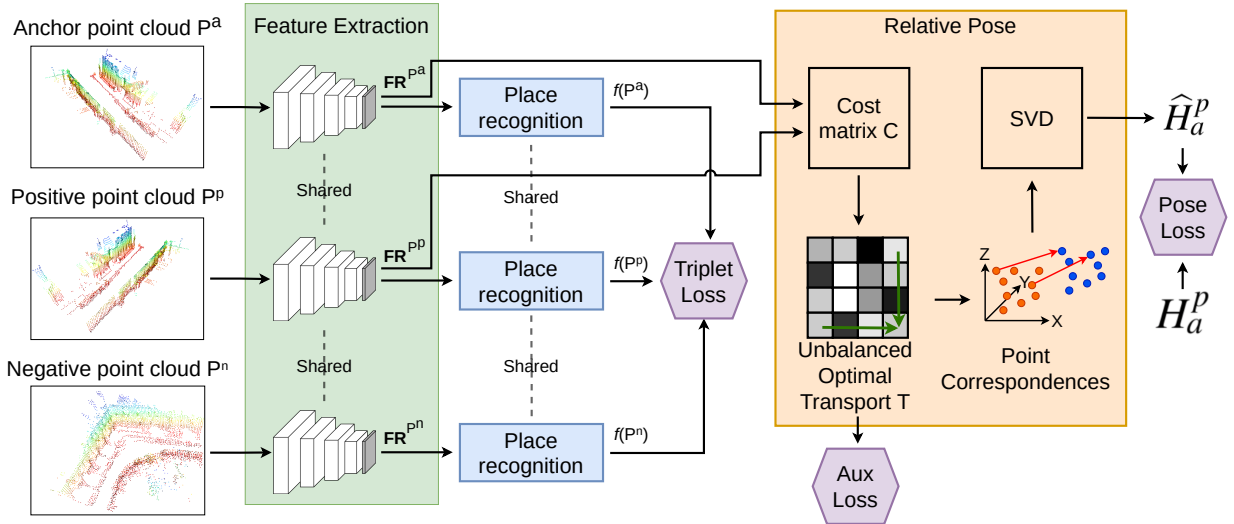


Figure 2. Overview of our proposed LCDNet which is composed of a shared feature extractor (green), a place recognition head (blue) that generates global descriptors, and a relative pose head (orange) that estimate the transformation between two point clouds. We use three loss functions to train LCDNet (triplet loss, aux loss, and pose loss) which are depicted in purple. The topology of the feature extractor is further illustrated in Figure 3.

is to employ a DNN for extracting features which are then used in the later stages to perform point clouds alignment. [20] propose such an approach known as PointNetLK, which exploits the PointNet [34] architecture for feature extraction and employs a variation of the Lucas and Kanade algorithm [40] to perform registration. Deep Closest Point (DCP) [23] is another approach that employs a Siamese architecture, attention modules, and differentiable Singular Value Decomposition (SVD) to regress a rigid transform for aligning two input point clouds. Recently, [21] proposes a DNN-based method called RPM-Net which is inspired by Robust Point Matching (RPM). RPM-Net employs two different neural networks to extract features and predict annealing parameters that are required for RPM. However, these methods are only capable of aligning point clouds that are relatively close to each other (up to 45° rotation misalignment), and completely fail to register point clouds that are more than 120° apart [20]. In contrast to the aforementioned methods, the approach that we propose in this work does not require any initial guess as input and can handle both outliers and occluded points. Moreover, unlike existing DNN-based methods, our approach effectively aligns point clouds with arbitrary initial rotation misalignment.

III. TECHNICAL APPROACH

In this section, we detail our proposed LCDNet for loop closure detection and point cloud registration from LiDAR point clouds. An overview of the proposed approach is depicted in Figure 2. The network consists of three main components: feature extraction, global descriptor head, and 6-DoF relative pose estimation head. We first describe each of the aforementioned components and the associated loss functions for training, followed by the approach for integrating LCDNet into the SLAM system.

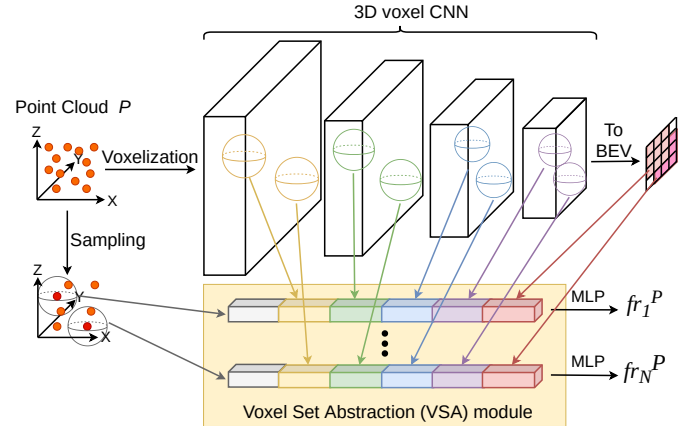


Figure 3. Network topology of the PV-RCNN architecture that we build upon for feature extractor component of our proposed LCDNet.

A. Feature Extraction

We build the feature extractor stream of our network based upon the PV-RCNN [41] architecture that was proposed for 3D object detection. PV-RCNN effectively combines the ability of voxel-based methods for extracting high-level features, with fine-grained features provided by PointNet-type architectures. We make several changes to the standard architecture to adapt it to our task. We illustrate the topology of our adapted PV-RCNN in Figure 3.

The input to the network is a point cloud $P \in \mathbb{R}^{J \times 4}$ (J points with 4 values each: x , y , z , and intensity). The output of our feature extractor network is a set of N keypoints' feature $\mathbf{FR}^P = \{fr_1^P, \dots, fr_N^P\}$, where $fr_i^P \in \mathbb{R}^D$ is the D -dimensional feature vector for the i -th keypoint. Since we are interested in the feature extraction, and not in the object detection head, we only use the 3D voxel DNN and the Voxel Set Abstraction (VSA) module, and we discarded the region proposal network, the ROI-grid pooling, and the fully connected layers towards the end of the architecture. The 3D voxel DNN first converts

the point cloud into a voxel grid of size $L \times W \times H$, where voxel features are averaged across all the points that lay within the same voxel. Subsequently, we extract a feature pyramid using sparse 3D convolutions and downsampling. In particular, we use four pyramid blocks composed of 3D sparse convolutions, with downsampling rates of $1\times$, $2\times$, $4\times$, and $8\times$, respectively. Finally, we convert the coarsest feature map into a 2D Bird's-Eye-View (BEV) feature map by stacking the features along the Z axis.

The VSA module, on the other hand, aggregates all the pyramid feature maps together with the BEV feature map and the input point cloud into a small set of N keypoints features. To do so, we first downsample the point cloud using the Farthest Point Sampling (FPS) algorithm [42] to select N uniformly distributed keypoints. The VSA module is an extension of the Set Abstraction (SA) level [43]. The standard SA aggregate neighbors point features in the raw point cloud, whereas, the VSA aggregate neighbors voxel features in the 3D sparse feature map. For every selected keypoint kp_i , and every layer l of the pyramid feature map, the keypoint features f_i^l are computed as

$$f_i^l = MP(MLP(\mathcal{M}(S_i^l))), \quad (1)$$

where MP is the max-pooling operation, MLP denotes a Multi Layer Perceptron (MLP), and \mathcal{M} randomly samples the set of neighbor voxel features S_i^l , which is computed as

$$S_i^l = \left\{ \left[fvox_j^l; v_j^l - kp_i \right]; \text{s. t. } \|v_j^l - kp_i\|^2 < r \right\}, \quad (2)$$

where $fvox_j^l$ is the feature of the voxel j at level l , v_j^l denotes the coordinates of the voxel j at level l , and r is the neighbor radius. This operation is performed at every level of the pyramid to yield

$$f_i^{pv} = [f_i^1, f_i^2, f_i^3, f_i^4]. \quad (3)$$

We perform a similar operation for the input raw point cloud, as well as the BEV feature map, yielding the aggregated keypoint features

$$f_i^{3D} = [f_i^{pv}, f_i^{raw}, f_i^{bev}]. \quad (4)$$

Lastly, we employ a MLP on the aggregated keypoint features to generate the final keypoint feature vectors as

$$fr_i = MLP(f_i^{3D}). \quad (5)$$

As opposed to the original PV-RCNN that processes only the points that lay in the camera Field Of View (FOV), we require the full 360° surrounding view. Therefore, we use a voxel grid size of $\pm 70.4\text{m}$, $\pm 70.4\text{m}$ and $[-1\text{m}, 3\text{m}]$ in the x, y and z dimensions, respectively. We use a voxel size of $0.1\text{m} \times 0.1\text{m} \times 0.1\text{m}$. We demonstrate the ability of our feature extractor in generating discriminative keypoint features by comparing it with different state-of-the-art backbones in the ablation studies presented in Section IV-F. Moreover, we also investigate the best choice for the dimensionality D of the keypoint features.

B. Global Descriptor

In order to generate a global descriptor for a given point cloud, we aggregate the keypoints' feature set \mathbf{FR}^P obtained from the feature extractor into a compact G -dimensional vector. To do so, we first employ the NetVLAD layer [27] which converts the $(N \times D)$ -dimensional \mathbf{FR}^P set into a $(K \times D)$ -dimensional vector $\mathbf{V}(\mathbf{FR}^P)$ by learning a set of K cluster centers $\{c_1, \dots, c_K\}$, $c_k \in \mathbb{R}^D$. NetVLAD mimics the original Vector of Locally Aggregated Descriptor (VLAD) [44] using differentiable operations. It replaces the k-means clustering with learnable clusters and replacing the hard assignment with a soft assignment defined as

$$a_k(fr_i^P) = \frac{e^{\mathbf{w}_k^\top fr_i^P + b_k}}{\sum_{k'=1}^K e^{\mathbf{w}_{k'}^\top fr_i^P + b_{k'}}}, \quad (6)$$

where $\mathbf{w}_k \in \mathbb{R}^D$ and $b_k \in \mathbb{R}$ are the learnable weights and bias. In practice, $a_k(fr_i^P)$ represents the probability of assigning the feature vector fr_i^P to the cluster center c_k . The final NetVLAD descriptor $\mathbf{V}(\mathbf{FR}^P) = [\mathbf{V}_1(\mathbf{FR}^P), \dots, \mathbf{V}_K(\mathbf{FR}^P)]$ is computed by combining the original VLAD formulation with the soft assignment defined in Equation (6) as

$$\mathbf{V}_k(\mathbf{FR}^P) = \sum_{i=1}^N a_k(fr_i^P)(fr_i^P - c_k). \quad (7)$$

We use the NetVLAD layer instead of max-pooling employed in PointNet [34], as it has demonstrated superior performance for point cloud retrieval [11]. To further reduce the dimensionality of the final global descriptor, we employ a simple MLP that compresses the $(K \times D)$ -dimensional vector $\mathbf{V}(\mathbf{FR}^P)$ into a G -dimensional compact descriptor. We then obtain the final global descriptor $f(P) \in \mathbb{R}^G$ by employing the Context Gating (CG) module [45] on the output of the MLP. The CG module re-weights the output of the MLP using a self-attention mechanism as

$$Y(X) = \sigma(WX + b) \odot X, \quad (8)$$

where X is the MLP output, σ is the element-wise sigmoid operation, \odot is the element-wise multiplication, W and b are the weights and bias of the MLP. The CG module captures dependencies among features by down-weighting or up-weighting features based on the *context* while considering the full set of features as a whole, thus focusing the attention on more discriminative features.

C. Relative Pose Estimation

Given two point clouds P and S , the third component of our architecture estimates the 6-DoF transformation to align the source point cloud P with the target point cloud S under driving conditions. We perform this task by matching the keypoints' features \mathbf{FR}^P and \mathbf{FR}^S computed using our feature extractor from Section III-A. Due to the sparse nature of LiDAR point clouds and the keypoint sampling step which is performed in the feature extractor, a point in P might not have a single matching point in S , but it can lay in between two or more

points in S . Therefore, a one-to-one mapping is not desirable in our task.

In order to address this problem, we employ the Sinkhorn algorithm [46], which can be used to approximate the *optimal transport* (OT) theory in a fast, highly parallelizable and differentiable manner. Recent work has shown benefits of using the Sinkhorn algorithm with DNNs for several tasks such as feature matching [47], scene flow [48], shape correspondence [49], and style transfer [50]. The discrete Kantorovich formulation of the optimal transport is defined as

$$T = \arg \min_{A \in \mathbb{R}^{N \times N}} \left\{ \sum_{i,j} C_{ij} A_{ij}; \text{ s. t. } A \text{ is doubly stochastic} \right\}, \quad (9)$$

where C_{ij} is the cost of matching the i -th point in P to the j -th point in S . In order to employ the Sinkhorn algorithm, we add an entropic regularization term:

$$T = \arg \min_{A \in \mathbb{R}^{N \times N}} \left\{ \sum_{i,j} C_{ij} A_{ij} + \lambda A_{ij} (\log A_{ij} - 1) \right\}, \quad (10)$$

where λ is a parameter that controls the sparseness of the mapping (as $\lambda \rightarrow 0$, T converges to a one-to-one mapping). However, both Equations (9) and (10) are subject to A being a doubly stochastic matrix (mass preservation constraint), *i.e.*, every point in \mathbf{FR}^P has to be matched to one or more points in \mathbf{FR}^S , and vice versa. In our point cloud matching task, some points in \mathbf{FR}^P might not have a matching in \mathbf{FR}^S , for example when a car is present in one point cloud but is absent in the other, or in the case of occlusions. Therefore, we need to relax the mass prevention constraint. One common approach to overcome this problem is by adding a dummy point in both P and S (*i.e.*, add a dummy row and column to A). Another way is to reformulate the problem as *unbalanced optimal transport* (UOT) which allows mass creation and destruction, and is defined as

$$T = \arg \min_{A \in \mathbb{R}^{N \times N}} \left\{ \left(\sum_{i,j} C_{ij} A_{ij} + \lambda A_{ij} (\log A_{ij} - 1) \right) + \rho \left(KL \left(\sum_i A_{ij} |U(1,N) \right) + KL \left(\sum_j A_{ij} |U(1,N) \right) \right) \right\}, \quad (11)$$

where KL is the Kullback–Leibler divergence, U is the discrete uniform distribution, and ρ is a parameter that controls how much mass is preserved. The UOT formulation, compared to the standard OT, reduces the negative effect caused by incorrect point matching and is more robust to the stochasticity induced by keypoint sampling [51]. A recent extension to the Sinkhorn algorithm [52] that approximates the unbalanced optimal transport is shown in Algo. 1. We set the cost matrix C as cosine distance between the keypoints' features $C_{ij} = 1 - FR_i^P \cdot FR_j^S / \|FR_i^P\| \|FR_j^S\|$. Instead of setting λ and ρ manually, we learn them using back propagation.

Once we estimate the unbalanced optimal transport T , which represents the set of soft correspondence between keypoints' features \mathbf{FR}^P and \mathbf{FR}^S , together with their respective 3D

Algorithm 1: Unbalanced Optimal Transport

Data: Cost matrix C , number of iterations L , parameters λ and ρ

Result: Unbalanced Optimal Transport T

begin

$K \leftarrow e^{-C/\lambda}$

$a \leftarrow \mathbb{1}_N/N$

$b \leftarrow \mathbb{1}_N/N$

$v \leftarrow \mathbb{1}_N/N$

for $i \leftarrow 1$ to L **do**

$u \leftarrow [a \odot (Kv)]^{\rho/(\rho+\lambda)}$

$v \leftarrow [b \odot (K^T u)]^{\rho/(\rho+\lambda)}$

end

$T \leftarrow u \odot K \odot v^T$

end

where \odot is the element wise division, and \odot is the element-wise multiplication.

keypoints' coordinates P and S , we compute for every keypoint $p_j \in P$ its projected coordinates in S as

$$\hat{s}_j = \frac{\sum_{k=1}^K T_{jk} s_k}{\sum_{k=1}^K T_{jk}}. \quad (12)$$

Finally, to estimate the rigid body transformation between the original point cloud P and its projection \hat{S} in S we use the weighted SVD. Since both Algo. 1 and SVD are differentiable, we train our relative pose head in an end-to-end manner by comparing the predicted transformation \hat{H}_P^S with the groundtruth transformation H_P^S .

Once the network has been trained, we replace the UOT-based relative position head with a RANSAC-based registration method that exploits the features extracted by our network to find correspondences. In this way, we can train the network in an end-to-end manner, and at the same time estimate accurate relative poses using the robust RANSAC estimator during inference.

D. Loss Function

We train our global descriptors using the triplet loss [53]. Given an anchor point cloud P^a , a positive sample P^p (point cloud of the same place), and a negative sample P^n (point cloud of a different place), the triplet loss enforces the distance between the descriptors of positive samples to be smaller than the distance between negative samples descriptors. More formally, the triplet loss is defined as

$$\mathcal{L}_{trp} = [d(f(P^a), f(P^p)) - d(f(P^a), f(P^n)) + m]_+, \quad (13)$$

where $d(\cdot)$ is a distance function, m is the desired separation margin, and $[x]_+$ means $\max(0, x)$.

Instead of selecting the triplets in advance (offline mining) for every anchor in the batch, we randomly select a positive sample, and we select the negative sample randomly from all the samples in the batch that depict a different place (online negative mining). We compute the relative pose transformation

only for positive pairs, and we train the model by comparing the anchor point cloud $P^a = \{p_1^a, \dots, p_j^a\}$ transformed using the predicted transformation \hat{H}_a^p and the groundtruth transformation H_a^p as

$$\mathcal{L}_{pose} = \frac{1}{J} \sum_{j=1}^J \left| \hat{H}_a^p p_j^a - H_a^p p_j^a \right|. \quad (14)$$

We add an auxiliary loss on the matches estimated by the unbalanced optimal transport T as

$$\mathcal{L}_{OT} = \frac{1}{J} \sum_{j=1}^J \left| \frac{\sum_{k=1}^K T_{jk} P_k^p}{\sum_{k=1}^K T_{jk}} - H_a^p p_j^a \right|. \quad (15)$$

The final loss function is a linear combination of the three aforementioned components:

$$\mathcal{L}_{total} = \mathcal{L}_{trp} + \mathcal{L}_{pose} + \beta \mathcal{L}_{OT}, \quad (16)$$

where β is loss balancing term which we empirically set to 0.05. Consequently, due to the combination of triplet loss, UOT, and data augmentation, the shared feature extractor learns to yield distinctive, rotation and translation invariant keypoints' features through backpropagation.

E. SLAM System

We integrate our proposed LCDNet into a recently proposed SLAM system, namely LIO-SAM [26] which achieves state-of-the-art performance on large-scale outdoor environments. LIO-SAM is a tightly coupled LiDAR inertial odometry framework built atop a factor graph. The framework takes a LiDAR point cloud and IMU measurements as input. It includes four types of constraints that are added to the factor graph: IMU preintegration, LiDAR odometry, GPS measurements (optional), and loop closure. In order to reduce the computational complexity, LIO-SAM selectively chooses LiDAR scans as keyframes only when the robot moves more than a predefined threshold since the last saved keyframe. The scans in between two keyframes are then discarded. We replaced the Euclidean distance-based loop closure detection provided in LIO-SAM with our LCDNet. From a technical perspective, for every keyframe \mathbb{F}_i added to the LIO-SAM factor graph, we compute and store its global descriptor $f(\mathbb{F}_i)$ in a database. When a new keyframe \mathbb{F}_{i+1} is added to the graph, we retrieve the point cloud with the most similar descriptor (excluding the past M keyframes) from the database:

$$W = \arg \min_{j \in \{1, \dots, i-M\}} \|f(\mathbb{F}_{i+1}) - f(\mathbb{F}_j)\|. \quad (17)$$

If the distance between the two descriptors is below a certain threshold th , we set \mathbb{F}_W as a loop candidate, and we estimate the 6-DoF transformation between the two point clouds \hat{H}_{i+1}^W provided by the relative pose head as described in Section III-C. Finally, we further refine the transformation using ICP with \hat{H}_{i+1}^W as initial guess, and we add the loop closure factor to the pose graph only if the ICP fitness score is higher than a threshold th_{icp} . By using this additional geometric consistency check, we can discard the few remaining false positive detection. It is important to note that no IMU nor GPS measurements are used in the loop detection step.

IV. EXPERIMENTAL EVALUATION

In this section, we first describe the datasets that we evaluate on, followed by the implementation details and the training protocol that we employ. We then present quantitative and qualitative results from experiments that are designed to demonstrate that our proposed LCDNet can (i) effectively detect loop closures even in challenging condition such as loops in the reverse direction, (ii) align two point clouds without any prior initial guess, (iii) robustly align point clouds that only partly overlap, (iv) provide an accurate initial guess for further ICP alignment, (v) integrate with an existing SLAM system to provide a fully featured localization and mapping framework, (vi) generalize to unseen environments.

A. Datasets

We evaluate our proposed approach on three different autonomous driving datasets. We detail the list of sequences that we use for training and testing, together with the respective number of loop closures and route direction of revisited places in Table I. Note that we do not include the sequences without loops.

KITTI: The KITTI odometry dataset [24] contains 11 sequences with LiDAR point clouds and groundtruth poses, six of which contain loops. However, the groundtruth for some of these sequences is not aligned to nearby loop closures. Therefore, we use the groundtruth provided with the SemanticKITTI dataset [54] which is consistent for all the sequences. Most of the KITTI odometry sequences contain loop closures from the same driving direction, except for sequence 08 which contains reverse loop closures. We evaluate our approach on sequences 00 and 08 as they contain the highest number of loops and reverse loops, respectively.

KITTI-360: The recently released KITTI-360 dataset [25] consists of nine sequences, six of which contain loops. KITTI-360 contains more loops and reverse loops than the standard KITTI dataset (see Table I). We evaluate our approach on two of the sequences in KITTI-360 that contain the highest number of loop closures: sequence 02 and sequence 09.

Freiburg: We recorded our own dataset by driving around the city of Freiburg, Germany, across different days. We used a car equipped with a Velodyne HDL-64E LiDAR sensor and an Applanix POS LV positioning system. The resulting dataset includes many loops, both from the same and reverse directions. Moreover, differently from the KITTI and KITTI-360 datasets, our Freiburg dataset includes many dynamic objects. The Freiburg dataset is thus used to evaluate the generalization ability of our approach to a different city, different sensor setup, and across different days by training the models on KITTI and KITTI-360, and evaluating them on our own dataset collected in Freiburg, without any re-training or fine-tuning.

B. Implementation and Training Details

Following [10], we consider two point clouds as a real loop if the distance between the groundtruth poses is less than four

Table I
STATISTICS OF EVALUATION DATASETS.

	KITTI						KITTI-360						Freiburg
	00	05	06	07	08	09	00	02	04	05	06	09	-
Num. of scans	4541	2761	1101	1101	4071	1591	10514	18235	11052	6291	9186	13247	25612
Num. of loops	790	492	69	97	334	18	2452	4690	2218	2008	2433	4670	13851
Num. of pairs	10499	6534	2138	2497	2960	252	24499	43894	21165	20361	22822	53858	~ 411M
Route direction	<i>Same</i>	<i>Same</i>	<i>Same</i>	<i>Same</i>	<i>Reverse</i>	<i>Same</i>	<i>Both</i>	<i>Both</i>	<i>Both</i>	<i>Both</i>	<i>Both</i>	<i>Both</i>	<i>Both</i>
% Reverse Loops	3%	5%	0%	0%	100%	0%	67%	87%	92%	88%	61%	46%	20%

meters. Moreover, we do not search for loop candidates in the past 50 scans to avoid detecting loops in nearby scans. We train LCDNet on sequences 05, 06, 07, and 09 of the KITTI dataset, validate it on sequences 00 and 08, and test it on the KITTI-360 dataset. We also train a second model, denoted as **LCDNet₊**, which is trained on sequences 00, 04, 05, and 06 of the KITTI-360 dataset, validated on sequences 02 and 09, and tested on the KITTI dataset.

We train all models for 150 epochs on a server with 4 NVIDIA TITAN RTX GPUs, using a batch size of 24 positive pairs. We use the ADAM optimizer to update the weights of the network, with an initial learning rate of 0.004 which is halved after epochs 40 and 80, and a weight decay of $5 \cdot 10^{-6}$. In all the experiments, if not otherwise specified, we set the number of keypoints $N = 4096$, the intermediate feature dimension $D = 640$, the output feature dimension $G = 256$, the number of NetVLAD clusters $K = 64$, the triplet margin $m = 0.5$, and the distance function in Equation (13) as the L2 distance. The number of iterations for the Sinkhorn algorithm is set to $L = 5$.

In order to help the network to learn viewpoint-invariant features, we apply a random rigid body transformation to each point cloud, with a maximum translation of $[\pm 1.5\text{m}]$ on the x and y axes, and $[\pm 0.25\text{m}]$ on the z axis; the maximum rotation of $[\pm 180^\circ]$ for the yaw (to simulate loop closures from different directions), and $[\pm 3^\circ]$ for roll and pitch.

C. Evaluation of Loop Closure Detection

To evaluate the loop closure detection performance of LCDNet, we use precision-recall curves and the Average Precision (AP) metric under two different evaluation protocols.

Protocol 1: In the first protocol, we evaluate our approach in a real loop closure setting. For each scan i of the sequence, we compute the similarity between the global descriptor $f(P^i)$ and the descriptor of all the previous scans, excluding the nearby scan as detailed in Section IV-A. We select scan j with the highest similarity as the loop candidate, and if the similarity between the two descriptors is higher than a threshold th , then we consider the pair (i, j) as a loop. In such a case, we further check the distance of the groundtruth poses between the two scans: if the distance is less than four meters, then we consider it as a true positive, and as a false positive otherwise. On the other hand, if the similarity is lower than the threshold, but if a scan within four meters around the current scan i exists, then we consider it as a false negative.

Protocol 2: In the second protocol, for each scan, we take into account all the previous scans, not only the one with the highest similarity. For every pair of scans, if the similarity between the two descriptors is higher than the threshold, we consider the pair as loop closure, and we compare against the groundtruth to compute precision and recall. Although in a real-world loop closure application only the most similar scan matters, if an approach is able to detect loops when the scans are very similar, but fails in more challenging scenarios (such as occlusions), this will not be reflected in the protocol 1 results. In protocol 2, on the other hand, all pairs of scans are considered, and thus approaches that better deal with challenging situations will achieve better results. Also in this protocol, we ignore nearby scans to avoid matching consecutive scans.

In both protocols, by varying the threshold th we obtain a set of pairs (precision, recall), that we use to generate the precision-recall curve and to compute the AP.

We compare our approach with state-of-the-art handcrafted methods: M2DP [9], Scan-Context [10], Intensity Scan-Context (ISC) [55], and LiDAR-IRIS [33], as well as DNN-based methods OverlapNet [13], and Semantic Graph Place Recognition (SG_PR) [14]. For all these approaches, we used the official code published by the respective authors, and the pretrained models that are provided by the authors for DNN-based methods. OverlapNet only provides the model trained with geometric information, we refer to this model as OverlapNet (*Geo*). All the DNN-based methods except for LCDNet₊ are trained on the KITTI dataset as described in Section IV-A, and evaluated individually on sequences from both KITTI and KITTI-360 datasets.

We present results with the AP metric for protocol 1 and protocol 2 in Table II. The best method is highlighted in bold, and the second best is underlined. Moreover, we present the precision-recall curves for both protocols in Figure 4. We observe that while most approaches achieve satisfactory results in detecting loop closures in the same direction (Figure 4 (a)), this is not the case for reverse loops as shown in Figure 4 (b). M2DP and SG_PR completely fail on the KITTI sequence 08; Scan Context, OverlapNet (*Geo*) and LiDAR-Iris also show a strong decrease in performance when dealing with reverse loops. For instance, the previous state-of-the-art method Scan Context achieved an AP of 0.96 in sequence 00 of the KITTI dataset (which contains only same direction loops), and 0.65 in sequence 08. Our proposed LCDNet, on the other hand, performs equally well for both reverse and same direction loops, achiev-

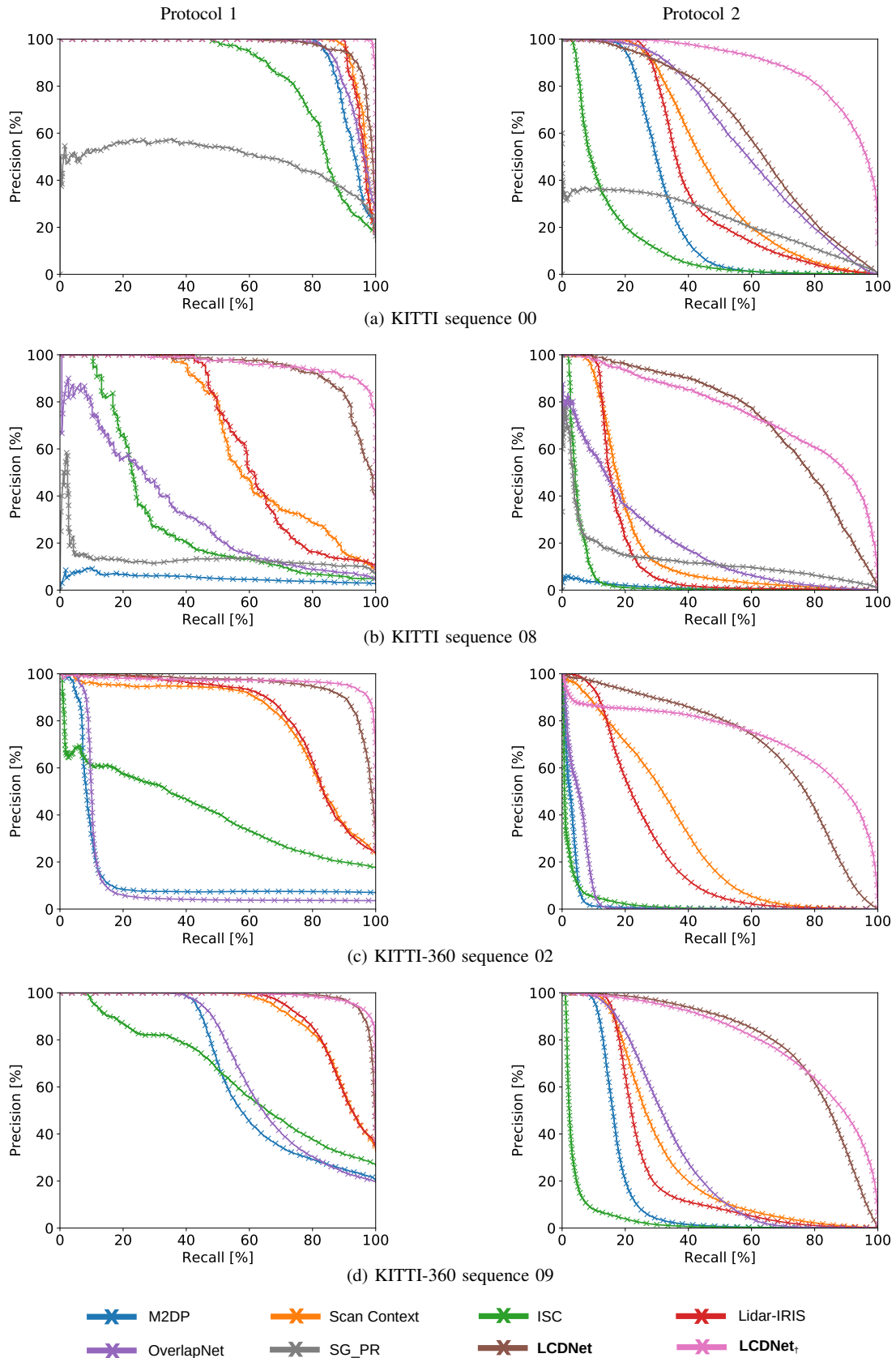


Figure 4. Comparison of loop closure detection precision-recall curves on KITTI (a-b) and on KITTI-360 (c-d) datasets evaluated using both protocols. Our proposed LCDNet₊ achieves the best performance in all the experiments, followed by our LCDNet as second best method. The improvement over previous state-of-the-art approaches is even more prominent when dealing with reverse direction loops, as observed in (b).

Table II

COMPARISON WITH THE STATE OF THE ART IN TERMS OF THE AVERAGE PRECISION EVALUATED ON THE KITTI AND KITTI-360 DATASETS.

Method		Protocol 1				Protocol 2			
		KITTI		KITTI-360		KITTI		KITTI-360	
		00	08	02	09	00	08	02	09
Handcrafted	M2DP [9]	0.93	0.05	0.15	0.66	0.31	0.01	0.03	0.17
	Scan Context [10]	0.96	0.65	0.81	0.90	0.47	0.21	0.32	0.31
	ISC [55]	0.83	0.31	0.41	0.65	0.14	0.05	0.03	0.04
	LiDAR-Iris [33]	0.96	0.64	0.83	0.91	0.42	0.17	0.25	0.26
DNN-based	OverlapNet [13]	0.95	0.32	0.14	0.70	0.60	0.20	0.05	0.33
	SG_PR [14]	0.49	0.13	-	-	0.23	0.13	-	-
	LCDNet	<u>0.97</u>	<u>0.94</u>	<u>0.95</u>	<u>0.98</u>	<u>0.62</u>	<u>0.73</u>	<u>0.69</u>	<u>0.79</u>
	LCDNet_‡	0.998	0.96	0.97	0.99	0.89	0.76	0.73	0.80

ing an AP of 0.94 and 0.97 respectively. This is even more noticeable in the results using protocol 2 where all the other approaches show a substantial decrease in performance, while our LCDNet achieves an AP score that is even better for detecting reverse loops than the same direction loops. We also observe that the model trained on the KITTI-360 dataset (LCDNet_‡) achieves the best performance on all the sequences, thereby setting the new state-of-the-art on both KITTI and KITTI-360.

D. Evaluation of Relative Pose Estimation

In this section, we evaluate the relative pose estimation between two point clouds. Our proposed LCDNet provides a full 6-DoF transformation under driving conditions between two points clouds. However, Scan Context, ISC, LiDAR-Iris, and OverlapNet only provide an estimation of the yaw angle. As M2DP, and SG_PR do not provide any information about the relative pose, we do not include them in the results presented in this section. Moreover, we compare our approach with state-of-the-art handcrafted methods for point cloud registration: ICP [16] using point-to-point and point-to-plane distances, RANSAC with FPFH features [37] and Fast Global Registration (FGR) [39], all implemented in the Open3D library [57], and TEASER++ [56] using the official implementation. We also compare with DNN-based methods RPMNet [21], Deep Closest Point (DCP) [23] and Product of Cross-Attention Matrices (PCAM) [22]. To provide a fair comparison, we trained all the latter DNN-based approaches on the same data, following the same protocol, and using the same number of keypoints used to train our LCDNet. Following [58], for the aforementioned handcrafted methods we first downsample the point clouds using a voxel size of 0.3 meter, while the latter DNN-based methods and our LCDNet perform point cloud registration using 4096 sampled points, which is a much sparser representation. Scan-Context, LiDAR-Iris, ISC, and OverlapNet, on the other hand, operate on spherical projections of the points, and thus they process almost all the points in the original cloud. We evaluate two versions of our method. The first one, denoted as *LCDNet (fast)*, leverages the output of the UOT-based relative position head to estimate the transformation. In the second version, denoted as *LCDNet*, we replace the UOT-based head with a RANSAC estimator, as described in Section III-C. The

models trained on KITTI-360 are denoted as *LCDNet_‡ (fast)* and *LCDNet_‡*, respectively. We also evaluate the performance of LCDNet followed by a further ICP registration. We report the latter evaluation only as a reference to show the best alignment achievable. Finally, we further investigate whether TEASER++ is a better pose estimator by replacing RANSAC in LCDNet.

We evaluate all the methods in terms of success rate (percentage of successfully aligned pairs), translation error (TE), and rotation error (RE) averaged over successful pairs as well as over all the positive pairs. We consider two pairs to be aligned successfully if the final rotation and translation error is below five degrees and two meters, respectively. The results on the KITTI and KITTI-360 datasets are reported in Tables III and IV. We observe that LiDAR-Iris achieves the best performance among the handcrafted methods and PCAM demonstrates superior results compared to existing DNN-based approaches when dealing with same and reverse direction pairs. However, as opposed to the other methods, PCAM only performs point cloud registration and do not provide any information regarding loop closure detection. Whereas, our proposed LCDNet and LCDNet_‡ achieve the highest success rates and lowest rotation errors compared to all the methods, with a success rate of 100% in three out of four sequences. PCAM, on the other hand, achieves the lowest translation errors in most sequences, but is not robust to registration under partial overlap, as we discuss in Section IV-E. The fast versions of our method achieve results comparable with, and in some sequences even better than existing approaches, while being much faster than most point cloud registration methods, as we discuss in Section IV-H. We observe that by replacing RANSAC in LCDNet and LCDNet_‡ with TEASER++ the success rates decrease and the translation errors significantly increase, while the rotation errors remain similar. During our experimental evaluations, we also observed that while the rotation and translation invariance obtained by our LCDNet primarily arise from our data augmentation scheme, many existing loop closure detection approaches (not reported in the comparison) did not converge at all when trained with the same scheme. Therefore, we argue that data augmentation by itself is not sufficient, and a well-designed architecture and loss function is necessary to achieve invariance.

Table III
COMPARISON OF RELATIVE POSE ERRORS (ROTATION AND TRANSLATION) BETWEEN POSITIVE PAIRS ON THE KITTI DATASET.

Approach	Seq. 00			Seq. 08			
	Success	TE [m] (succ. / all)	RE [deg] (succ. / all)	Success	TE [m] (succ. / all)	RE [deg] (succ. / all)	
Handcrafted	Scan Context* [10]	97.66%	- / -	1.34 / 1.92	98.21%	- / -	1.71 / 3.11
	ISC* [55]	32.07%	- / -	1.39 / 2.13	81.28%	- / -	2.07 / 6.27
	LiDAR-Iris* [33]	98.83%	- / -	0.65 / 1.69	<u>99.29%</u>	- / -	0.93 / 1.84
	ICP (P2p) [16]	35.57%	0.97 / 2.08	1.36 / 8.98	0%	- / 2.43	- / 160.46
	ICP (P2pl) [16]	35.54%	1.00 / 2.11	1.39 / 8.99	0%	- / 2.44	- / 160.45
	RANSAC [37]	33.95%	0.98 / 2.75	1.37 / 12.01	15.61%	1.33 / 4.57	1.79 / 37.31
	FGR [39]	34.54%	0.98 / 5972.31	1.2 / 12.79	17.16%	1.32 / 35109.13	1.76 / 28.98
	TEASER++ [56]	34.06%	0.98 / 2.72	1.33 / 15.85	17.13%	1.34 / 3.83	1.93 / 29.19
DNN-based	OverlapNet* [13]	83.86%	- / -	1.28 / 3.89	0.10%	- / -	2.03 / 65.45
	RPMNet [21]	47.31%	1.05 / 2.07	0.60 / 1.88	27.80%	1.28 / 2.42	1.77 / 13.13
	DCP [23]	50.71%	0.98 / 1.83	1.14 / 6.61	0%	- / 4.01	- / 161.24
	PCAM [22]	99.68%	0.07 / 0.08	0.35 / 0.74	94.90%	0.19 / 0.41	0.51 / 6.01
Ours	LCDNet (fast)	93.03%	0.65 / 0.77	0.86 / 1.07	60.71%	1.02 / 1.62	1.65 / 3.13
	LCDNet	100%	<u>0.11 / 0.11</u>	0.12 / 0.12	100%	0.15 / 0.15	0.34 / 0.34
	LCDNet _‡ (fast)	<u>99.79%</u>	0.28 / 0.29	0.30 / 0.30	88.51%	0.66 / 0.93	1.00 / 1.31
	LCDNet _‡	100%	0.14 / 0.14	<u>0.14 / 0.14</u>	100%	<u>0.18 / 0.18</u>	<u>0.36 / 0.36</u>
LCDNet + ICP	100%	0.04 / 0.04	0.09 / 0.09	100%	0.09 / 0.09	0.33 / 0.33	
LCDNet _‡ + ICP	100%	0.04 / 0.04	0.08 / 0.08	100%	0.07 / 0.07	0.32 / 0.32	
LCDNet + TEASER	94.39%	0.66 / 0.77	0.09 / 0.10	71.99%	1.05 / 1.62	0.33 / 0.35	
LCDNet _‡ + TEASER	99.78%	0.28 / 0.29	0.09 / 0.09	89.39%	0.67 / 0.93	0.33 / 0.34	

* these approaches only estimate the rotation between two point clouds, therefore are not directly comparable with the other approaches which estimate the full 6-DoF transformation under driving conditions.

Table IV
COMPARISON OF RELATIVE POSE ERRORS (ROTATION AND TRANSLATION) BETWEEN POSITIVE PAIRS ON THE KITTI-360 DATASET.

Approach	Seq. 02			Seq. 09			
	Success	TE [m] (succ. / all)	RE [deg] (succ. / all)	Success	TE [m] (succ. / all)	RE [deg] (succ. / all)	
Handcrafted	Scan Context* [10]	92.31%	- / -	1.60 / 5.49	95.25%	- / -	1.40 / 6.80
	ISC* [55]	83.15%	- / -	1.71 / 3.44	86.26%	- / -	1.51 / 7.08
	LiDAR-Iris* [33]	96.54%	- / -	1.07 / 2.24	97.63%	- / -	0.72 / 3.80
	ICP (P2p) [16]	4.19%	1.10 / 2.26	1.74 / 149.76	21.24%	1.06 / 2.22	1.34 / 66.34
	ICP (P2pl) [16]	4.19%	1.11 / 2.30	1.18 / 149.39	21.29%	1.07 / 2.24	1.38 / 66.23
	RANSAC [37]	24.78%	1.24 / 3.67	1.83 / 32.22	29.69%	1.12 / 3.14	1.48 / 23.42
	FGR [39]	27.92%	1.23 / 6758.87	1.85 / 18.16	30.46%	1.12 / 6011.39	1.44 / 17.35
	TEASER++ [56]	27.02%	1.25 / 3.16	1.83 / 19.16	30.32%	1.14 / 2.91	1.46 / 19.22
DNN-based	OverlapNet* [13]	11.42%	- / -	1.79 / 76.74	54.33%	- / -	1.38 / 33.62
	RPMNet [21]	37.99%	1.18 / 2.26	1.30 / 5.97	41.42%	1.13 / 2.21	1.02 / 3.95
	DCP [23]	5.62%	1.09 / 3.14	1.36 / 149.27	30.10%	1.04 / 2.30	1.06 / 64.86
	PCAM [22]	97.46%	0.20 / 0.30	0.75 / 1.36	<u>99.78%</u>	0.12 / 0.13	0.51 / 0.64
Ours	LCDNet (fast)	83.92%	0.84 / 1.10	1.28 / 1.67	89.49%	0.76 / 0.94	0.99 / 1.19
	LCDNet	98.62%	0.28 / 0.32	<u>0.32 / 0.35</u>	100%	<u>0.18 / 0.18</u>	0.20 / 0.20
	LCDNet _‡ (fast)	89.07%	0.40 / 0.45	0.57 / 0.62	98.87%	0.43 / 0.44	0.59 / 0.63
	LCDNet _‡	<u>98.55%</u>	<u>0.27 / 0.32</u>	0.32 / 0.34	100%	0.20 / 0.20	<u>0.22 / 0.22</u>
LCDNet + ICP	98.51%	0.20 / 0.25	0.24 / 0.27	100%	0.10 / 0.10	0.15 / 0.15	
LCDNet _‡ + ICP	98.51%	0.20 / 0.25	0.24 / 0.27	100%	0.11 / 0.11	0.15 / 0.15	
LCDNet + TEASER	86.63%	0.85 / 1.10	0.40 / 0.52	90.57%	0.76 / 0.94	0.22 / 0.25	
LCDNet _‡ + TEASER	98.06%	0.40 / 0.45	0.37 / 0.45	99.10%	0.43 / 0.44	0.22 / 0.23	

* these approaches only estimate the rotation between two point clouds, therefore are not directly comparable with the other approaches which estimate the full 6-DoF transformation under driving conditions.

E. Partial Overlap

In this section, we evaluate the ability of LCDNet in detecting loops and regressing the relative pose between point clouds that only overlap partially. To do so, we follow the same evaluation protocol that we use in Section IV-C (protocol 1) and Section IV-D. We simulate partial overlapping pairs by removing a random section of each point cloud. We compare LCDNet against state-of-the-art approaches on the sequence 08 of the KITTI dataset under two settings: by removing a random 45° and 90° sector, respectively. Table V reports the results of this experiment in terms of average precision (AP), success rate, mean translation error and mean rotation error. Although the AP of LCDNet drops moderately when a 90° section is removed, LCDNet_† still achieves an AP higher than all the existing approaches evaluated on the complete overlap test (Table II). We observe that PCAM which achieves remarkable results in the full overlap registration test, struggles when dealing with partial overlapping point clouds with a success rate that drops from 95% to 56%, a translation error that increases from 0.41 m to 3.32 m, and a rotation error that raises from 6.01° to 34.64° . LCDNet and LCDNet_†, on the other hand, retain an almost perfect success rate and slightly lower translation and rotation errors.

We also investigated the MulRan dataset [59] for this experiment, as the LiDAR mounted on their vehicle is obstructed by the radar sensor for approximately 70° rear FOV. Therefore, in reverse direction scenarios, the scans share only a very limited overlap. In preliminary evaluations, all the considered approaches failed in detecting reverse loops. We argue that this is a limitation of all scan-to-scan methods, and that scan-to-map approaches should be considered in these scenarios.

F. Ablation Studies

In this section, we present ablation studies on the different architectural components of our proposed LCDNet. All the models presented in this section are trained on the KITTI dataset, and evaluated on the sequence 08 using the AP, mean rotation error (RE) and mean translation error (TE) metrics. We choose sequence 08 as the validation set since it is the most challenging sequence, containing only reverse direction loops. Since RANSAC does not influence the training of the network, in this section the rotation and translation errors are computed using the *LCDNet (fast)* version.

We first compare our feature extractor built upon PVRCNN presented in Section III-A with three different backbones: the widely adopted feature extractor PointNet [34], the dynamic graph CNN EdgeConv [35], and the recent state-of-the-art semantic segmentation network RandLA-Net [60]. We modified all backbones in order to output a feature vector of size $D = 640$ for $N = 4096$ points, similar to our backbone. We report results in Table VI. The ability of our feature extractor presented in Section III-A to combine high-level features from the 3D voxel DNN with fine-grained details provided by the PointNet-based voxel set abstraction layer is demonstrated by the superior performance compared to other backbones, outperforming them

in every metric by a large margin. Our backbone built upon PV-RCNN achieves an average precision of 0.94 compared to 0.67 achieved by the second best backbone. For relative pose estimation, PV-RCNN achieves a mean rotation error of 3.13° and a mean translation error of 1.62 m compared to 16.85° achieved by EdgeConv and 3.55 m achieved by RandLA-Net.

In Table VII, we present ablation studies on the architecture of the relative pose head, the dimensionality of the extracted point features, the effect of the auxiliary optimal transport loss presented in Equation (15), and the number of keypoints. We first compare our UOT-based relative pose head presented in Section III-C with a MLP that directly regresses the rotation and translation, similar to [12]. In particular, we train three models using different rotation representations. The first model, *MLP(sin-cos)* uses two parameters to represent the rotation: the sine and cosine of the yaw angle. *MLP(quat)* represents the rotation as unit quaternions, and *MLP(bingham)* uses the Bingham representation proposed in [61]. From the first set of rows in Table VII, we observe that our proposed relative pose head significantly outperforms the MLP-based heads, especially in the rotation estimation. Our proposed relative pose head achieves a mean rotation error of 3.13° compared to 21.05° achieved by the best MLP model. Moreover, the UOT-based head favors keypoint features that are rotation and translation invariant, thus enabling the backbone to learn more discriminative features, consequently also improving the loop closure detection performance. The MLP based heads, on the other hand, require rotation specific features in order to predict the transformation, which hinders the performance of the place recognition head, which can be observed from the lower average precision achieved by these models.

Subsequently, we study the influence of the dimensionality of point features on the performance of our approach. We train four models by varying dimensionality D as 640, 128, 64, and 32. From the results shown in the second set of rows in Table VII, we observe that the performances decrease with lowering the dimensionality D .

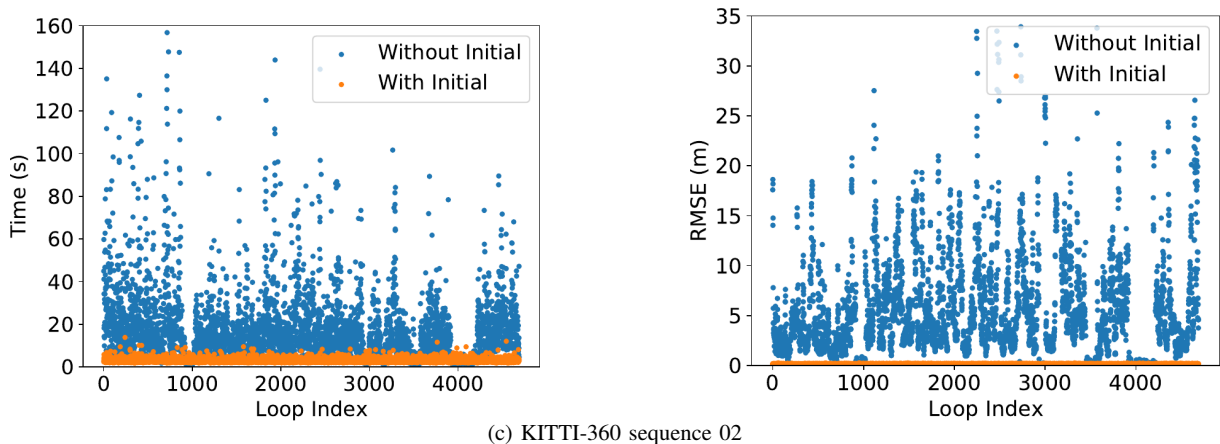
We evaluate the performance of LCDNet without the auxiliary loss presented in Equation (15). From the results shown in the third set of rows of Table VII, we observe that when training without the optimal transport loss, the performance in terms of average precision and relative transformation decreases significantly. This demonstrates that the auxiliary optimal transport loss enables the network to learn more distinctive features, which benefits the performance of both loop closure detection and relative transformation estimation.

Finally, in the last set of rows of Table VII we compare the performance of LCDNet to changes in the number of selected keypoints N . Predictably, the performances increase with adding more keypoints. However, the average precision does not improve when increasing the number of keypoints to 8192. Therefore, due to the higher memory and computation required, we use 4096 keypoints in our final model.

Table V
COMPARISON OF LOOP CLOSURE DETECTION (AP) AND RELATIVE POSE ERRORS (ROTATION AND TRANSLATION) UNDER PARTIAL OVERLAP ON THE SEQUENCE 08 OF THE KITTI DATASET.

Approach		45°				90°			
		AP	Success	TE [m] (all)	RE [deg] (all)	AP	Success	TE [m] (all)	RE [deg] (all)
Handcrafted	Scan Context* [10]	0.52	27.33%	-	57.70	0.40	17.40%	-	72.05
	LiDAR-Iris* [33]	0.43	97.84%	-	2.78	0.22	96.28%	-	5.13
	ICP (P2p) [16]	-	0%	2.42	160.46	-	0%	2.42	160.46
	ICP (P2pl) [16]	-	0%	2.45	160.46	-	0%	2.45	160.42
	RANSAC [37]	-	15.51%	4.88	43.77	-	13.78%	5.50	48.74
	FGR [39]	-	16.55%	44439.37	30.30	-	14.49%	235332.54	34.20
	TEASER++ [56]	-	16.42%	4.03	30.32	-	15.98%	4.37	34.99
DNN-based	OverlapNet* [13]	0.09	1.11%	-	70.69	0.01	0.68%	-	85.68
	PCAM [22]	-	84.67%	1.04	11.80	-	55.62%	3.32	34.64
Ours	LCDNet	<u>0.79</u>	100%	<u>0.20</u>	<u>0.38</u>	<u>0.59</u>	<u>99.93%</u>	<u>0.24</u>	<u>0.46</u>
	LCDNet _†	0.83	100%	0.19	0.36	0.70	100%	0.21	0.37

* these approaches only estimate the rotation between two point clouds, therefore are not directly comparable with the other approaches which estimate the full 6-DoF transformation under driving conditions.



(c) KITTI-360 sequence 02

Figure 5. Comparison of time (left) and RMSE (right) between ICP without initial guess and ICP with the LCDNet prediction as the initial guess on the sequence 02 of the KITTI-360 dataset. Results on other sequences show similar behaviour, and are thus not reported for brevity. The initial guess provided by our LCDNet significantly reduces both runtime and final error on sequences containing reverse loops.

Table VI
ABLATION STUDY ON THE BACKBONE NETWORK ARCHITECTURE.

Backbone	AP	TE [m]	RE [deg]
PointNet [34]	<u>0.67</u>	5.15	34.14
EdgeConv [35]	0.52	5.44	<u>16.85</u>
RandLA-Net [60]	0.55	<u>3.55</u>	20.08
PVRCNN [41]	0.94	1.62	3.13

G. ICP with Initial Guess

In this experiment, we evaluate the performance of employing LCDNet as an initial guess for further refinement using ICP. We compare the runtime and the final Root Mean Square Error (RMSE) of ICP without any initial guess and ICP with LCDNet relative pose estimate as an initial guess. The time of ICP with initial guess also includes the network inference time. Results from this experiment are presented in Figure 5 and two qualitative results are shown in Figure 6. While only dealing with the same direction loops, ICP achieves satisfactory results and the initial guess does not improve the performance

significantly. However, when reverse loops are present, ICP often fails in accurately registering the two point clouds. In this case, the initial guess from our LCDNet greatly reduces both the runtime and final errors of ICP as observed in Figure 5.

From Figure 6, we see that ICP fails when the rotation misalignment between the two point clouds is significant. On the other hand, LCDNet accurately aligns these two point clouds and it improves the results even further while using ICP with LCDNet prediction as initial guess. On average, ICP with LCDNet initial guess is 4 times faster than ICP without any initial guess and achieves an RMSE which is 22 times lower. Note that in the results presented in Figure 5, we use the whole point clouds to perform the registration with ICP.

H. Runtime Analysis

In this section, we compare the runtime of LCDNet with existing state-of-the-art approaches for loop detection. All experiments were performed on a system with an Intel i7-6850K CPU and an NVIDIA GTX 1080 ti GPU. We use the

Table VII

ABLATION STUDY ON THE DIFFERENT ARCHITECTURAL COMPONENTS OF OUR LCDNET EVALUATED ON SEQUENCE 08 OF THE KITTI DATASET.

Relative Pose Head	Feature Size D	Auxiliary Loss	Num Keypoints	AP	TE [m]	RE [deg]
UOT				0.94	<u>1.62</u>	<u>3.13</u>
MLP (sin-cos)	640	✓	4096	0.75	2.14	21.05
MLP (quat)				0.78	2.43	35.16
MLP (bingham)				0.75	2.27	22.69
UOT	640			0.94	<u>1.62</u>	<u>3.13</u>
	128	✓	4096	<u>0.92</u>	1.85	3.19
	64			<u>0.92</u>	1.99	3.33
	32			0.86	2.23	4.09
UOT	640	✓	4096	0.94	<u>1.62</u>	<u>3.13</u>
		✗		0.83	6.00	4.71
UOT	640	✓	8192	0.94	1.28	1.99
			4096	0.94	<u>1.62</u>	<u>3.13</u>
			2048	0.85	4.68	3.73
			1024	0.69	5.17	4.75
			512	0.50	4.79	4.85

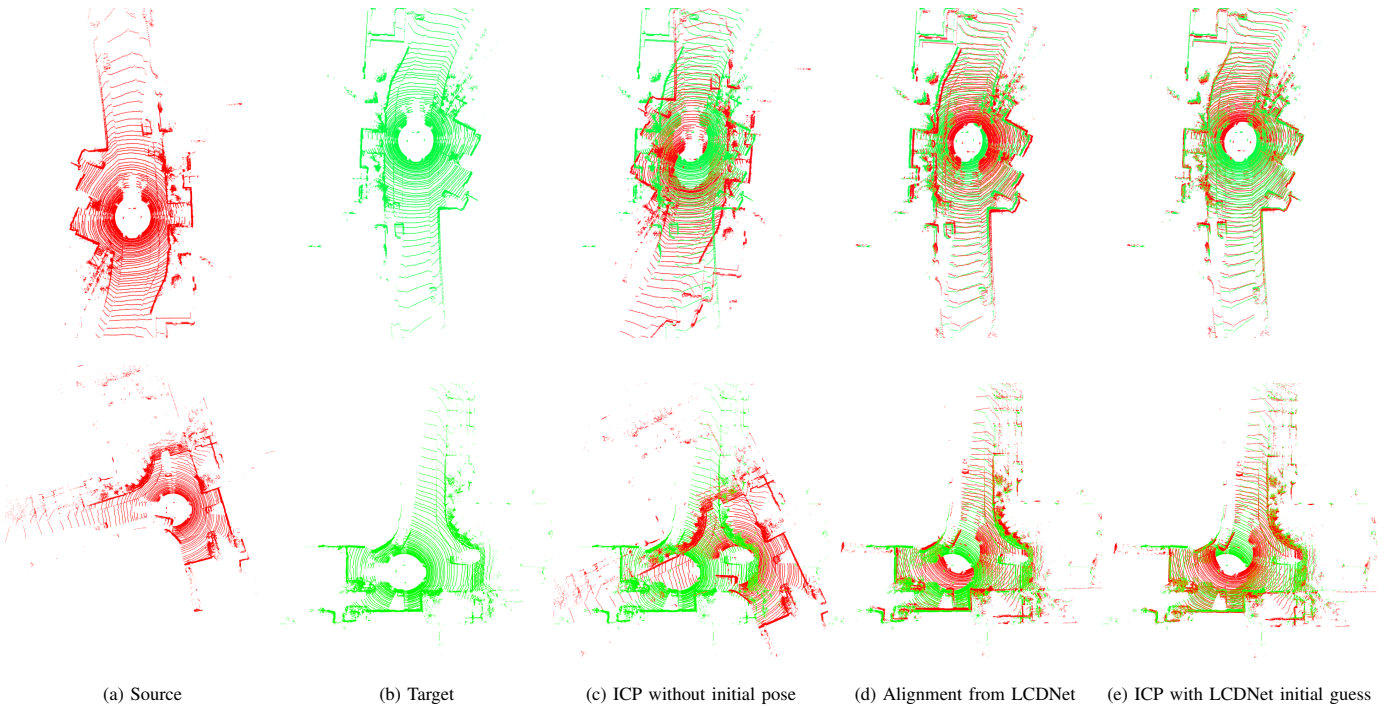


Figure 6. Qualitative comparison of ICP alignment with and without using the LCDNet prediction as an initial guess. ICP alone (c) is not able to register the source (a) and the target (b) when the initial rotation misalignment is high. Whereas, our LCDNet effectively aligns them (d). The final ICP alignment with the prediction of our LCDNet as the initial guess further improve the results (e).

Table VIII

COMPARISON OF RUNTIME ANALYSIS FOR THE LOOP CLOSURE TASK.

Method	Descriptor Extraction [ms]	Pairwise Comparison [ms]	Map Querying [ms]	GPU Required
M2DP [9]	169.28	0.01	5	✗
SC [10]	3.66	0.11	2000	✗
SC-50 [10]	3.66	0.11	6.96	✗
ISC [55]	1.97	0.53	9000	✗
LiDAR-Iris [33]	8.13	5.39	98000	✗
OverlapNet [13]	16.00	6.00	109000	✓
LCDNet	94.60	0.01	5	✓

official implementation of existing approaches as described in Sections IV-C and IV-D. Results from this experiment are presented in Table VIII in which the descriptor extraction time also includes the preprocessing required by the respective method. The pairwise comparison represents the time required to compare the descriptors of two point clouds. In the map querying column, we report the time for comparing the descriptor of one scan with that of all the previous scans in the KITTI-360 sequence 02, which amounts to 18 235 comparisons in total. For methods that do not require an ad-hoc function to compare descriptors (LCDNet and M2DP), we use the efficient FAISS library [62] for similarity search in order to

Table IX
COMPARISON OF RUNTIME ANALYSIS FOR THE POINT CLOUD
REGISTRATION TASK.

	Method	Descriptor Extract. [ms]	Pairwise Reg. [ms]	Total [ms]	GPU
Handcrafted	SC [10]	3.66	0.11	7.43	✗
	ISC [55]	1.97	0.53	4.47	✗
	LiDAR-Iris [33]	8.13	5.39	21.65	✗
	ICP (P2p) [16]	-	25.53	25.53	✗
	ICP (P2pl) [16]	8.16	35.83	52.15	✗
	RANSAC [37]	24.99	299.66	349.64	✗
	FGR [39]	24.99	188.74	238.72	✗
	TEASER++ [56]	24.99	94.89	144.87	✗
DNN-based	OverlapNet [13]	16.00	6.00	38.00	✓
	RPMNet [21]	366.75	121.29	854.79	✓
	DCP [23]	19.56	78.76	117.88	✓
	PCAM [22]	187.71	80.77	456.18	✓
Ours	LCDNet (fast)	94.60	4.70	193.9	✓
	LCDNet	94.60	1135	1324.2	✓

build and query the map. Scan Context also introduces the *ring key* descriptors which enable fast search for finding loop candidates, at the expense of detection performances. We also report the runtime of scan context using the ring key, denoted as *Scan Context-50*. However, it is important to note that the results reported in IV-C were computed without the ring key.

As shown in Table VIII, the methods that require an ad-hoc comparison function (ISC, LiDAR-Iris, and OverlapNet) are not suited for real-time applications, since they require up to 100 seconds to perform a single query. Whereas, LCDNet queries more than 18000 scans in five milliseconds. Although it is possible to further reduce the time required to query the map when integrating the loop closure approaches in a SLAM system, such as using the covariance-based radius search [13], in this experiment we evaluate the runtime in the case where no prior information about the current pose is available.

We report the runtime for aligning two point clouds by LCDNet and existing approaches in Table IX. For methods that perform both loop closure and point cloud alignment (SC, ISC, LiDAR-IRIS, OverlapNet, and LCDNet), the descriptor extraction time is shared between the two tasks. While LCDNet (fast) is faster than most DNN-based approaches for point cloud registration (RPMNet and PCAM), LCDNet is slightly slower than RPMNet. On the other hand, some approaches are much faster than both LCDNet and LCDNet (fast); however, they either only estimate a 1-DoF transformation (SC, ISC, and LiDAR-Iris), or achieve unsatisfactory performances (ICP, RANSAC, FGR, TEASER++, and DCP). It is important to note that the point cloud registration task does not need to run in realtime, since it is only required after a loop closure is detected. Moreover, LCDNet is the only method that performs both loop closure detection and 6-DoF point cloud registration under driving conditions.

I. Qualitative Results

We present the qualitative results from LCDNet and LCDNet_‡ on sequences from both the KITTI and KITTI-360

Table X
COMPARISON WITH THE STATE OF THE ART ON DATA FROM THE
GENERALIZATION EXPERIMENTS IN FREIBURG.

	Method	AP
Handcrafted	M2DP [9]	0.60
	Scan Context [10]	0.74
	ISC [55]	0.38
	LiDAR-Iris [33]	0.73
DNN-based	OverlapNet [13]	0.59
	LCDNet	<u>0.79</u>
	LCDNet_‡	0.88

datasets in Figure 7. We show the true positive, false positive, and false negative scans overlaid with the respective groundtruth trajectories. We observe that while LCDNet effectively detects same direction and reverse direction loops, it also fails to detect some loops (false negative) and detects some loops where there should be no loops (false positive). LCDNet_‡ further improves the performance by reducing the number of false positives and false negatives, while still maintaining accurate true positive detections. On the KITTI sequence 08, LCDNet yields some false negative detections that are almost completely eliminated by LCDNet_‡, although few false positive scans are still detected. On sequence 02 of the KITTI-360 dataset, LCDNet presents a large amount of false negatives which are significantly reduced by LCDNet_‡. Similarly, on the KITTI-360 sequence 09, LCDNet presents a few false positive detections that are completely eliminated by LCDNet_‡.

J. Evaluation of Complete SLAM System

We integrate our proposed approach into LIO-SAM [26], which is a recent state-of-the-art LiDAR SLAM system, by replacing its loop closure detection pipeline with LCDNet. We evaluate the entire SLAM system on the sequence 02 of the KITTI dataset. In particular, we evaluate LIO-SAM integrated with LCDNet and we compare it with the original LIO-SAM. We observed that our approach detects loop closures where the original LIO-SAM fails to do so due to the presence of the accumulated drift. In Figure 8, we report the results obtained with both SLAM systems and show the distance error between LIO-SAM keyframes and groundtruth poses. We can observe that the high error (red) associated with the path of the original LIO-SAM is caused by the failed loop closure detection, since the system drifts significantly along the z-dimension. Conversely, LCDNet detects such loops, perform the closure and improve the overall performance of LIO-SAM.

We publicly release the integrated LIO-SAM system with our LCDNet at <http://rl.uni-freiburg.de/research/lidar-slam-lc>.

K. Generalization Analysis

Finally, in this section, we evaluate the generalization ability of our proposed LCDNet by analyzing the performance in unseen environments and on different robot platforms. We evaluate both LCDNet and LCDNet_‡ in real-world experiments in Freiburg using a car with a rack of LiDAR sensors mounted

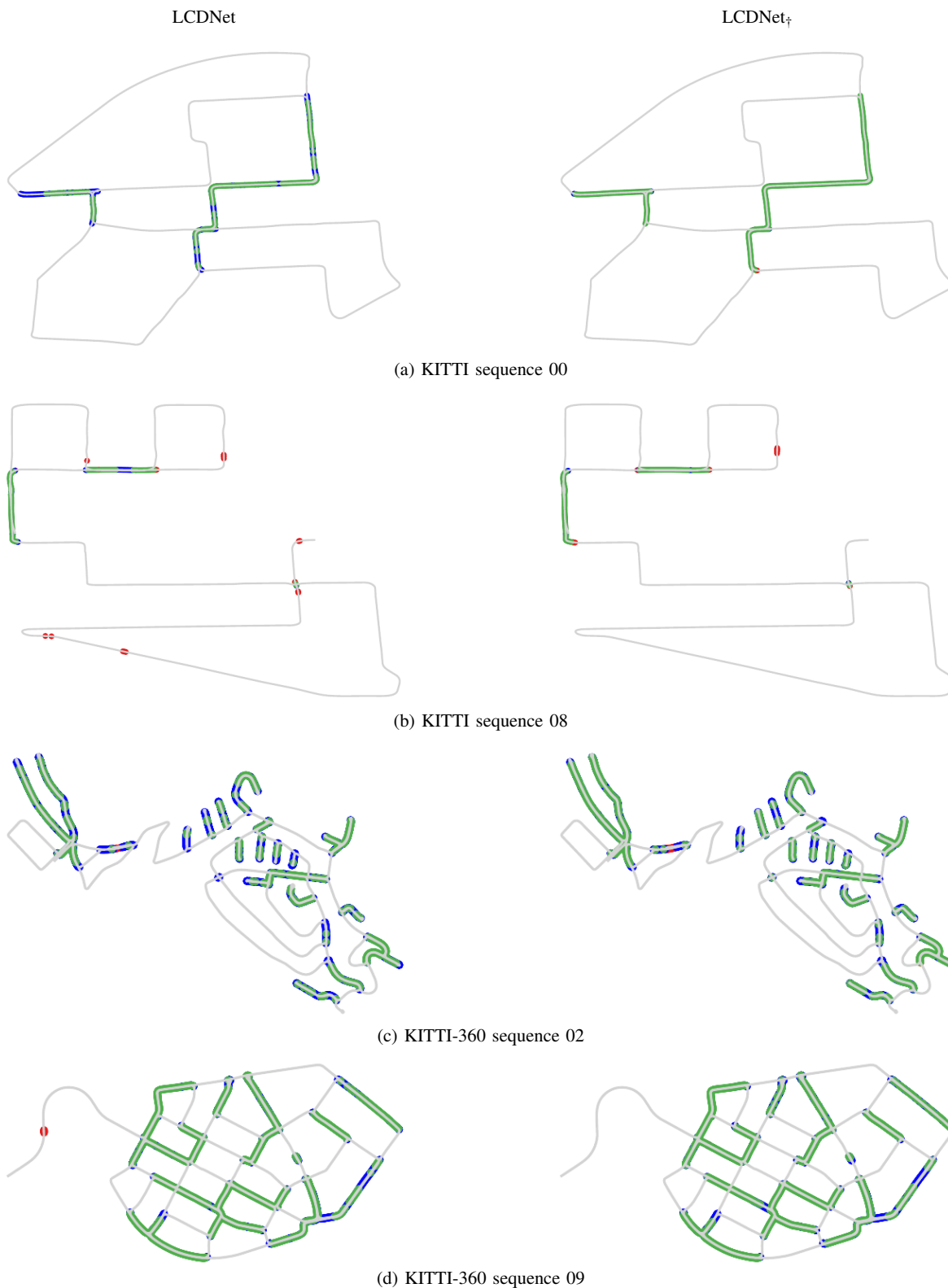


Figure 7. Qualitative loop closure detection results of LCDNet on KITTI (a-b) and KITTI-360 (c-d) datasets. Green points ● are true positive detections, red points ● are false positive, and blue points ● are false negative. The left column shows results of LCDNet trained on the KITTI dataset, while the right columns shows results of LCDNet₇ trained on the KITTI-360 dataset. While both LCDNet and LCDNet₇ effectively detects loops in all the sequences, LCDNet₇ further reduces the number of false positive and false negative detections.

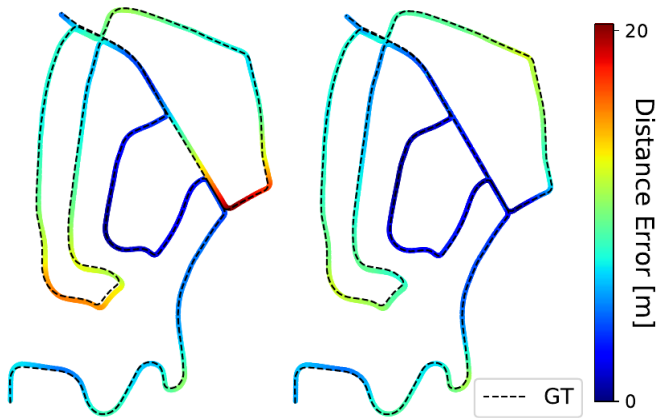


Figure 8. Performance of LIO-SAM with the original loop closure detection method (left) compared to our approach (right) on sequence 02 of the KITTI dataset.

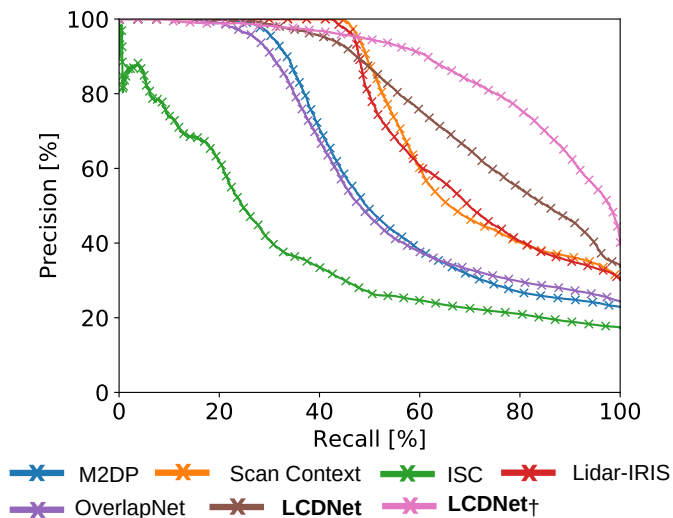


Figure 9. Comparison of precision-recall curves evaluated using protocol 1 on data from the generalization experiments in Freiburg.

on the roof as shown in Figure 10 (bottom right). Note that in these experiments, we do not retrain or fine-tune LCDNet and LCDNet₊ on any data from Freiburg. The KITTI and KITTI-360 datasets on which we trained our models on, were primarily recorded in narrow roads, but the streets of Freiburg also include dual carriageways, therefore we increase the range at which two scans are considered to be a real loop from 4 to 10 meters. In Figure 9, we compare the precision-recall curves of our approach with handcrafted and DNN-based methods using protocol 1 (Section IV-C). We do not report results using protocol 2 for this experiment as there are more than 400 million positive pairs in this trajectory.

As shown in Table X, Scan Context achieves the best performance among the handcrafted methods, with an AP of 0.74. Nevertheless, our LCDNet and LCDNet₊ outperform all the other approaches achieving an AP of 0.79 and 0.88, respectively. Since the data from Freiburg consists of many reverse loops, existing approaches often fail to detect them, leading to a decrease in their performance. Our approach

demonstrates exceptional performance even though it has never seen scans from Freiburg during training. Moreover, we employ our modified version of LIO-SAM to generate the trajectory and the map of the experimental runs in Freiburg. In Figure 10, we show the resulting map overlaid on the aerial image. The results show that the map is well aligned with the aerial image and there is no evidence of any drift. This demonstrates that our LCDNet effectively corrects the accumulated drift. It is important to note that the precision-recall curve and the AP of our LCDNet are computed based only on the global descriptor extracted by the place recognition head. However, in the modified SLAM system, we additionally perform a consistency check (Section III-E) based on the transformation predicted by the relative pose head which further discard the remaining false positive detections.

Finally, we also exploit the Freiburg dataset to demonstrate the point cloud alignment ability of our approach in new environments. Since we do not have an accurate pose for each LiDAR frame, we generate groundtruth transformations using the GPS poses and ICP, and discarding pairs that produce an inaccurate alignment. First, for each frame we identify possible pairs by considering its neighbors within a distance of 10m. In order to avoid the pairs that are composed of consecutive frames, given a point cloud we discard the previous and the following $n = 100$ frames. Secondly, for each pair we compute the yaw angle difference Δ_{yaw} and define three difficulty levels. Then, for each frame we select a random pair for every category whenever possible. We set the maximum number of ICP iterations $n_{icp} = 1000$, and we only consider pairs with a fitness score $fit \geq 0.6$ and an inlier correspondences $rmse \leq 0.3m$. Finally, we randomly sample the resulting pairs to have about the same number of same direction and reverse direction loops. The resulting number of pairs amounts to 4246, of which 2106 are reverse loops.

The results reported in Table XI show that our LCDNet effectively generalizes to new environments for the point cloud registration task. LCDNet and LCDNet₊ achieve a success rate of 98.94% and 99.81%, respectively, compared to the second best method that achieves 92.49%. The overall mean translation error of LCDNet₊ is more than two times smaller, and the rotation error is an order of magnitude lower than PCAM.

V. CONCLUSIONS

In this paper, we presented the novel LCDNet architecture for loop closure detection and point cloud registration. LCDNet is composed of a shared feature extractor built upon the PV-RCNN network, a place recognition head that captures discriminative global descriptors, and a novel differentiable relative pose head based on the unbalanced optimal transport theory which effectively aligns two point clouds without any prior information regarding their initial misalignment. We identified a discrepancy in the evaluation protocols of existing methods, therefore we performed uniform evaluations of state-of-the-art handcrafted as well as DNN-based loop closure detection methods.

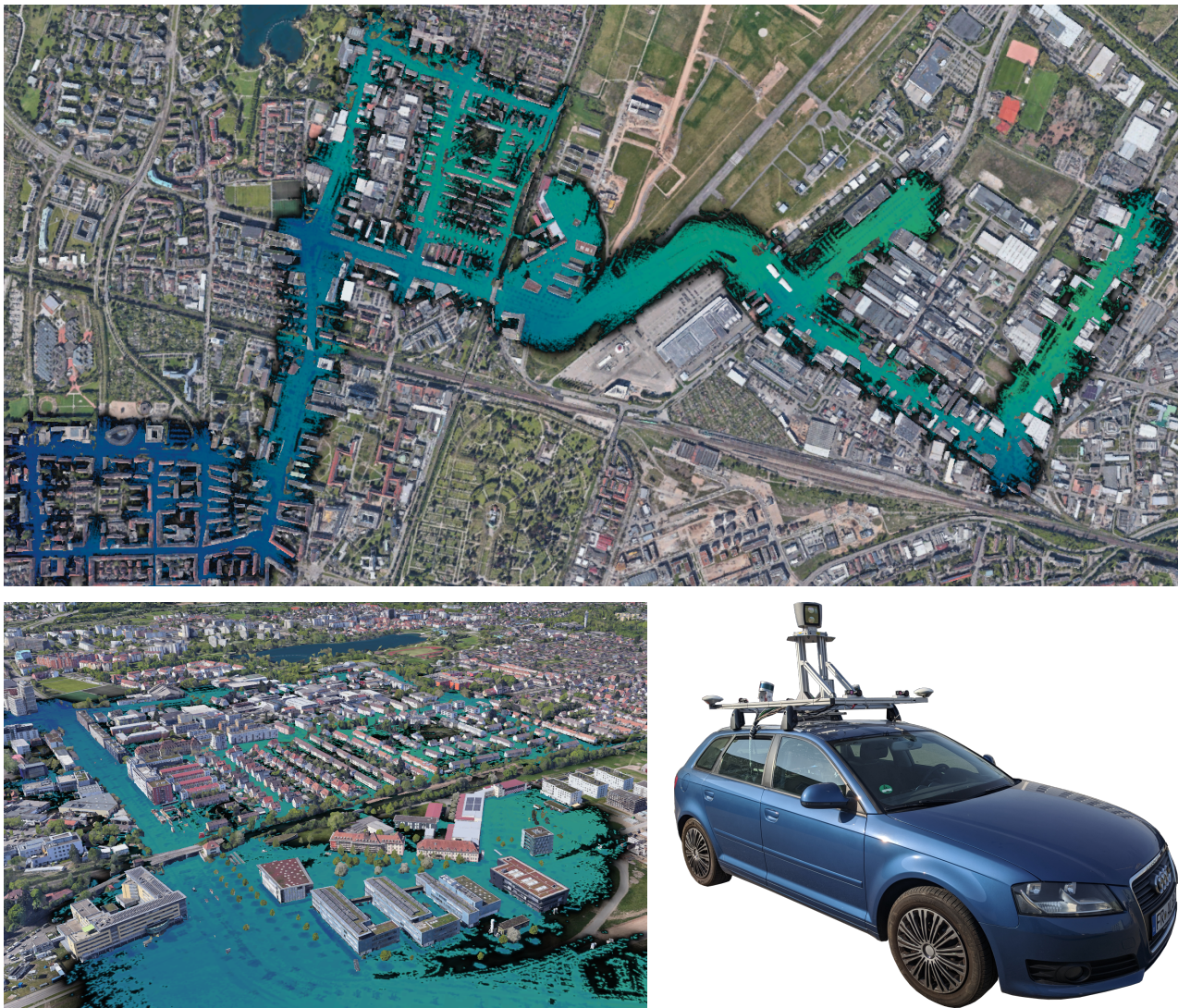


Figure 10. Qualitative results of our approach on data from the generalization experiments in Freiburg. The final map generated from LIO-SAM integrated with our LCDNet is overlaid on the georeferenced aerial images. Image on the top shows the entire map, while the images in the bottom show zoomed in segments and the car used to collect the dataset. The color of the point cloud is based on the Z-coordinates of the points from lowest (blue) to highest (green).

We presented extensive evaluations of LCDNet on the KITTI odometry and KITTI-360 datasets, which demonstrates that our approach sets the new state-of-the-art and successfully detects loops even in challenging conditions such as reverse direction loops, where existing methods fail. Our LCDNet₊ achieves an average precision of 0.96 on the sequence 08 of the KITTI dataset which contains only reverse direction loops, compared to 0.65 AP of the previous state-of-the-art method. Our proposed relative pose head demonstrates impressive results, outperforming existing approaches for point cloud registration and loop closure detection as well as different heads based on the standard MLPs. Our LCDNet aligns opposite direction point clouds with an average rotation error of 0.34° , and 0.15 m for the translation components, compared to 1.84° and 0.41 m achieved by LiDAR-IRIS and PCAM, respectively. Moreover, LCDNet is robust to partial overlapping point cloud, retaining a 100% success rate when removing a 90° sector from each point cloud, while the second best method drop from 95%

to 55%. We also showed that the relative pose prediction from our approach can further be refined using ICP for accurate registration. We integrated our LCDNet with LIO-SAM to provide a complete SLAM system which can detect loops even in presence of strong drift. Additionally, we demonstrated the generalization ability of our approach by evaluating it on the data from experiments using a different robotic platform and in an unseen city from that which was used for training. Finally, we have made the code and the SLAM system publicly available to encourage research in this direction.

ACKNOWLEDGMENTS

This work was partly funded by the Federal Ministry of Education and Research (BMBF) of Germany under SORTIE, and by the Eva Mayr-Stihl Stiftung. The authors would like to thank Johan Vertens for his assistance in the data collection.

Table XI
COMPARISON OF RELATIVE POSE ERRORS (ROTATION AND
TRANSLATION) BETWEEN POSITIVE PAIRS ON THE FREIBURG DATASET.

	Approach	Success	TE [m] (succ. / all)	RE [deg] (succ. / all)
Handcrafted	Scan Context [10]	59.30%	- / -	1.36 / 52.70
	ISC [55]	55.51%	- / -	1.52 / 51.02
	LiDAR-Iris [33]	69.95%	- / -	1.52 / 51.02
	ICP (P2p) [16]	29.06%	0.83 / 2.60	1.21 / 89.79
	ICP (P2pl) [16]	28.73%	0.88 / 2.62	1.22 / 89.83
	RANSAC [37]	29.96%	1.01 / 3.54	1.34 / 31.29
	FGR [39]	27.72%	0.97 / 313.258	1.27 / 13.46
	TEASER++ [56]	29.49%	0.99 / 3.37	1.31 / 11.94
DNN-based	OverlapNet [13]	42.79%	- / -	1.31 / 70.91
	RPMNet [21]	32.05%	0.87 / 2.57	1.09 / 46.99
	DCP [23]	12.25%	1.26 / 5.22	1.19 / 87.04
	PCAM [22]	92.49%	0.40 / 0.67	0.50 / 4.28
Ours	LCDNet	98.94%	0.39 / 0.42	0.32 / 0.37
	LCDNet _†	99.81%	0.28 / 0.28	0.18 / 0.18
	LCDNet + ICP	99.79%	0.22 / 0.23	0.16 / 0.16
	LCDNet _† + ICP	99.86%	0.21 / 0.22	0.14 / 0.14
	LCDNet + TEASER	33.61%	1.09 / 3.74	0.17 / 0.42
	LCDNet _† + TEASER	33.37%	1.15 / 4.45	0.13 / 0.27

REFERENCES

- [1] D. Cattaneo, D. G. Sorrenti, and A. Valada, "Cmrnet++: Map and camera agnostic monocular visual localization in lidar maps," *Int. Conf. on Robotics & Automation Workshop on Emerging Learning and Algorithmic Methods for Data Association in Robotics*, 2020.
- [2] M. Mittal, R. Mohan, W. Burgard, and A. Valada, "Vision-based autonomous uav navigation and landing for urban search and rescue," *Int. Symposium of Robotics Research*, 2019.
- [3] N. F. Tanke, G. A. Long, D. Agrawal, A. Valada, G. A. Kantor, et al., "Automation of hydroponic installations using a robot with position based visual feedback," in *Proceedings of the international conference of agricultural engineering CIGR-Ageng*, 2012.
- [4] A. Valada, P. Velagapudi, B. Kannan, C. Tomaszewski, G. Kantor, and P. Scerri, "Development of a low cost multi-robot autonomous marine surface platform," in *Field and service robotics*, 2014, pp. 643–658.
- [5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [6] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *Int. Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [7] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard, "Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation," in *Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 1249–1255.
- [8] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," in *Int. Conf. on Robotics & Automation*, 2013.
- [9] 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.
- [10] 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*, pp. 1–19, 2021.
- [11] M. Angelina Uy and G. Hee Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.
- [12] L. Schaupp, M. Bürki, R. Dubé, R. Siegwart, and C. Cadena, "Oreos: Oriented recognition of 3d point clouds in outdoor scenarios," in *Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 3255–3261.
- [13] 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.
- [14] X. Kong, X. Yang, G. Zhai, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3d point clouds," *Int. Conf. on Intelligent Robots and Systems*, 2020.
- [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 *Int. Conf. on Intelligent Robots and Systems*, 2020.
- [16] 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.
- [17] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems*, 2009.
- [18] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2241–2254, 2015.
- [19] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti, "Point clouds registration with probabilistic data association," in *Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 4092–4098.
- [20] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.
- [21] Z. J. Yew and G. H. Lee, "RPM-Net: Robust Point Matching Using Learned Features," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 11 824–11 833.
- [22] A.-Q. Cao, G. Puy, A. Boulch, and R. Marlet, "PCAM: Product of Cross-Attention Matrices for Rigid Registration of Point Clouds," in *Int. Conf. on Computer Vision*, 2021.
- [23] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Int. Conf. on Computer Vision*, 2019, pp. 3523–3532.
- [24] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.
- [25] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3d to 2d label transfer," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [26] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 5135–5142.
- [27] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.
- [28] X. Zhang, Y. Su, and X. Zhu, "Loop closure detection for visual slam systems using convolutional neural network," in *International Conference on Automation and Computing*, 2017, pp. 1–6.
- [29] Q. Liu and F. Duan, "Loop closure detection using cnn words," *Intelligent Service Robotics*, vol. 12, no. 4, pp. 303–318, 2019.
- [30] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3d surf for robust three dimensional classification," in *Europ. Conf. on Computer Vision*, 2010, pp. 589–602.
- [31] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Int. Conf. on Computer Vision Workshops*, 2009, pp. 689–696.
- [32] S. Siva, Z. Nahman, and H. Zhang, "Voxel-based representation learning for place recognition based on 3d point clouds," in *Int. Conf. on Intelligent Robots and Systems*, 2020.
- [33] Y. Wang, Z. Sun, J. Yang, and H. Kong, "Lidar iris for loop-closure detection," in *Int. Conf. on Intelligent Robots and Systems*, 2020.
- [34] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

- [35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, 2019.
- [36] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: a graph theoretic approach," in *Int. Conf. on Robotics & Automation*, 2000, pp. 2512–2517.
- [37] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Int. Conf. on Robotics & Automation*, 2009, pp. 3212–3217.
- [38] 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.
- [39] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Europ. Conf. on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016, pp. 766–782.
- [40] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, p. 674–679.
- [41] 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 *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 10529–10538.
- [42] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical computer science*, vol. 38, pp. 293–306, 1985.
- [43] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.
- [44] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.
- [45] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint arXiv:1706.06905*, 2017.
- [46] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The annals of mathematical statistics*, vol. 35, no. 2, pp. 876–879, 1964.
- [47] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 4937–4946.
- [48] G. Puy, A. Boulch, and R. Marlet, "FLOT: Scene Flow on Point Clouds Guided by Optimal Transport," in *Europ. Conf. on Computer Vision*, 2020, pp. 527–544.
- [49] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers, "Deep Shells: Unsupervised Shape Correspondence with Optimal Transport," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [50] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style Transfer by Relaxed Optimal Transport and Self-Similarity," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 10043–10052.
- [51] K. Fatras, T. Séjourné, R. Flamary, and N. Courty, "Unbalanced minibatch optimal transport; applications to domain adaptation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3186–3197.
- [52] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Scaling algorithms for unbalanced optimal transport problems," *Mathematics of Computation*, vol. 87, no. 314, pp. 2563–2609, 2018.
- [53] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015.
- [54] 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.
- [55] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, 2020.
- [56] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
- [57] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.
- [58] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "Mulran: Multimodal range dataset for urban place recognition," in *Int. Conf. on Robotics & Automation*, May 2020.
- [59] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 11108–11117.
- [60] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly, "A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty," in *Robotics: Science and Systems*, 2020.
- [61] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.



Daniele Cattaneo received the M.S. degree in Computer Science from the University of Milano-Bicocca, Milan, Italy, in 2016 and the Ph.D. degree in Computer Science from the same university in 2020. He is currently a Postdoctoral Researcher with the Robotic Learning Lab of the University of Freiburg, Freiburg, Germany, headed by Abhinav Valada. His research interest includes deep learning for robotic perception and state estimation, with a focus on sensor fusion, cross-modal matching, and domain generalization.



Matteo Vaghi received the B.S. and M.S. degrees in Computer Science from the University of Milano-Bicocca, Milan, Italy, in 2016 and 2019, respectively. He was a junior research assistant at the IRALab Research Group at the same university in 2019 and 2020. Currently, He is a Ph.D. student at the University of Milano-Bicocca and his research focuses on the development of techniques for addressing the vehicle localization problem in urban areas. In particular, his main research topics are computer vision, deep learning and robotics.



Abhinav Valada is an Assistant Professor and Director of the Robot Learning Lab at the University of Freiburg. He is a member of the Department of Computer Science, a principal investigator at the BrainLinks-BrainTools Center, and a founding faculty of the European Laboratory for Learning and Intelligent Systems (ELLIS) unit at Freiburg. He received his Ph.D. in Computer Science from the University of Freiburg in 2019 and his M.S. degree in Robotics from Carnegie Mellon University in 2013. His research lies at the intersection of robotics, machine learning and computer vision with a focus on tackling fundamental robot perception, state estimation and control problems using learning approaches in order to enable robots to reliably operate in complex and diverse domains. Abhinav Valada is a Scholar of the ELLIS Society, a DFG Emmy Noether Fellow, and co-chair of the IEEE RAS TC on Robot Learning.