SGLoc: Scene Geometry Encoding for Outdoor LiDAR Localization

Wen Li[1,2*] Shangshu Yu[1,2*] Cheng Wang[1,2†] Guosheng Hu[3] Siqi Shen[1,2] Chenglu Wen[1,2]
[1] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China
[2] Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, School of Informatics, Xiamen University, China
[3] Oosto, Belfast, UK

Abstract

LiDAR-based absolute pose regression estimates the global pose through a deep network in an end-to-end manner, achieving impressive results in learning-based localization. However, the accuracy of existing methods still has room to improve due to the difficulty of effectively encoding the scene geometry and the unsatisfactory quality of the data. In this work, we propose a novel LiDAR localization framework, SGLoc, which decouples the pose estimation to point cloud correspondence regression and pose estimation via this correspondence. This decoupling effectively encodes the scene geometry because the decoupled correspondence regression step greatly preserves the scene geometry, leading to significant performance improvement. Apart from this decoupling, we also design a tri-scale spatial feature aggregation module and inter-geometric consistency constraint loss to effectively capture scene geometry. Moreover, we empirically find that the ground truth might be noisy due to GPS/INS measuring errors, greatly reducing the pose estimation performance. Thus, we propose a pose quality evaluation and enhancement method to measure and correct the ground truth pose. Extensive experiments on the Oxford Radar RobotCar and NCLT datasets demonstrate the effectiveness of SGLoc, which outperforms state-of-the-art regression-based localization methods by 68.5% and 67.6% on position accuracy, respectively.

1. Introduction

Estimating the position and orientation of LiDAR from point clouds is a fundamental component of many applications in computer vision, *e.g.*, autonomous driving, virtual reality, and augmented reality.

Contemporary state-of-the-art LiDAR-based localization methods explicitly use maps, which match the query point cloud with a pre-built 3D map [18, 23, 27, 49]. However, these methods usually require expensive 3D map storage and communication. One alternative is the regression-based approach, absolute pose regression (APR), which directly estimates the poses in the inference stage without maps [8, 24, 25, 40, 45]. APR methods typically use a CNN to encode the scene feature and a multi-layer perceptron to regress the pose. Compared to map-based methods, APR does not need to store the pre-built maps, accordingly reducing communications.

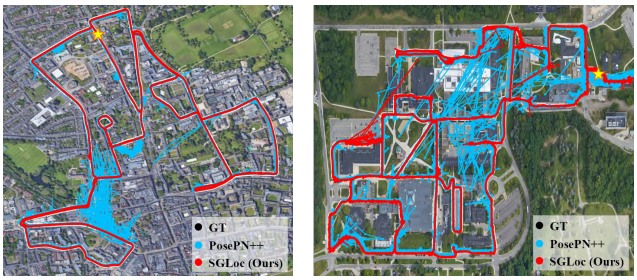

Figure 1. LiDAR Localization results of our method and PosePN++ [51] (state-of-the-art method) in urban (left) and school (right) scenes from Oxford Radar RobotCar [2] and NCLT [34] datasets. The star indicates the starting position.

For (1), APR networks learn highly abstract global scene representations, which allow the network to classify the scene effectively [25]. However, the global features usually cannot encode detailed scene geometry, which is the key to achieving an accurate pose estimation [10, 11, 38, 39]. Prior efforts have tried to minimize the relative pose or photometric errors to add geometry constraints by pose graph optimization (PGO) [4, 21] or novel view synthesis (NVS) [10, 11]. However, this introduces additional computations, limiting its wide applications. For (2), we empirically find current large-scale outdoor datasets suffer from various errors in the data due to GPS/INS measuring errors. It affects the APR learning process and makes it difficult to evaluate the localization results accurately. To our knowledge, the impact of data quality on localization has not been carefully investigated in the existing literature.

*Equal contribution.
†Corresponding author.

This paper proposes a novel framework, SGLoc, which can (1) effectively capture the scene geometry; In addition, we propose a data pre-processing method, Pose Quality Evaluation and Enhancement (PQEE), which can (2) improve data quality. (1) Existing APR methods conduct end-to-end regression from the point cloud in LiDAR coordinate to pose. Unlike them, SGLoc decouples this process to (a) regression from the point cloud in LiDAR coordinate to world coordinate and (b) pose estimation via the point cloud correspondence in LiDAR and world coordinate using RANSAC [17]. Importantly, step (a) can effectively preserve the scene geometry, which is key for pose estimation [10, 11, 38, 39]. To achieve high accuracy in step (a), we design a Tri-scale Spatial Feature Aggregation (TSFA) module and an Inter-Geometric Consistency Constraint (IGCC) loss to effectively capture scene geometry. (2) We empirically find that pose errors in the data greatly degrade the pose estimation performance. For example, the ground truth pose obtained by GPS/INS suffers from measuring errors. To address this problem, we proposed a PQEE method which can measure the errors in the pose and correct them afterward. We conduct extensive experiments on Oxford Radar RobotCar [2] and NCLT [34] datasets, and results show that our method has great advantages over the state-of-the-art, as demonstrated in Fig. 1.

Our contributions can be summarized as follows:

- SGLoc is the first work to decouple LiDAR localization into point cloud correspondences regression and pose estimation via predicted correspondences, which can effectively capture scene geometry, leading to significant performance improvement.
- We propose a novel Tri-Scale Spatial Feature Aggregation (TSFA) module and an Inter-Geometric Consistency Constraint (IGCC) loss to further improve the encoding of scene geometry.
- We propose a generalized pose quality evaluation and enhancement (PQEE) method to measure and correct the pose errors in the localization data, improving 34.2%/16.8% on position and orientation for existing LiDAR localization methods.
- Extensive experiments demonstrate the effectiveness of SGLoc, which outperforms state-of-the-art LiDAR localization methods by 68.1% on position accuracy. In addition, to our knowledge, we are the first to reduce the error to the level of the sub-meter on some trajectories.

2. Related Work

2.1. Map-based Localization

Map-based methods aim to match the query point cloud with a pre-built 3D map, which can be classified into retrieval-based [18, 27, 30, 44, 49] and registration-based methods [14, 16, 20, 32, 37, 47]. Retrieval-based methods cast localization as a place recognition problem, which finds the most similar point cloud in the database. Registration-based localization performs fine matching between the query point cloud and the pred-built map. DCP [47] is an impressive work investigating geometry correspondences generated by weighted calculation for pose estimation. Unlike this, SGLoc directly regresses the correspondences. In addition, Some methods combine both retrieval and registration techniques to achieve better localization accuracy in dense urban areas [9, 12, 50, 52]. However, the high cost of 3D map storage and communication limits the widespread application of map-based methods.

2.2. Regression-based Localization

Recently, in camera localization, the scene coordinate regression method has been designed, which regresses 2D-3D correspondences and estimates the pose with PnP-RANSAC [4, 6, 7, 22, 54]. While these methods have shown impressive results in small and medium-sized scenes, *e.g.*, 7-Scenes [42], they do not scale well to large scenes [5]. HSCNet attempts to solve this problem by conditioning discrete location labels around each pixel [28], but it still cannot be applied to large-scale outdoor scenes, *e.g.*, the street scene in the Cambridge landmark dataset [25], which covers an area of about $5\mathrm{hm}^2$.

Absolute pose regression. APR methods typically use the same basic pipeline: first, extracting high-level features using a CNN and then using these features to regress the 6-DoF pose. PoseNet [25] originally defines this task using a modified GoogleNet [43] to regress camera poses. Successors of PoseNet have focused on improving the framework through model architecture and loss function. E-PoseNet [33] proposes a translation and rotation equivariant CNN that directly induces representations of camera motions into the feature space. PAE [41] uses the pose auto-encoders framework to significantly reduce model parameters. However, the above methods learn highly abstract global scene representations, leading to various wrong predictions due to the lack of effective scene geometry encoding.

Recent research suggests scene geometry is key to accurate pose estimation [10, 11, 38, 39]. Geometric PoseNet [24] investigates geometric loss for learning to regress position and orientation simultaneously with scene geometry. MapNet [8] adds pairwise geometric constraints between video frames using additional VO algorithms. AtLoc [45] utilizes a self-attention mechanism to adaptively focus on the import targets in the scene. MS-Transformer [40] proposes a transformer-based method to learn robust features. Direct-PoseNet [11] adapts additional photometric loss by comparing the query image with NVS on the predicted pose. DFNet [10] explores a direct matching scheme in feature space, leading to a more robust performance than Direct-PoseNet. However, these methods require auxiliary algo-

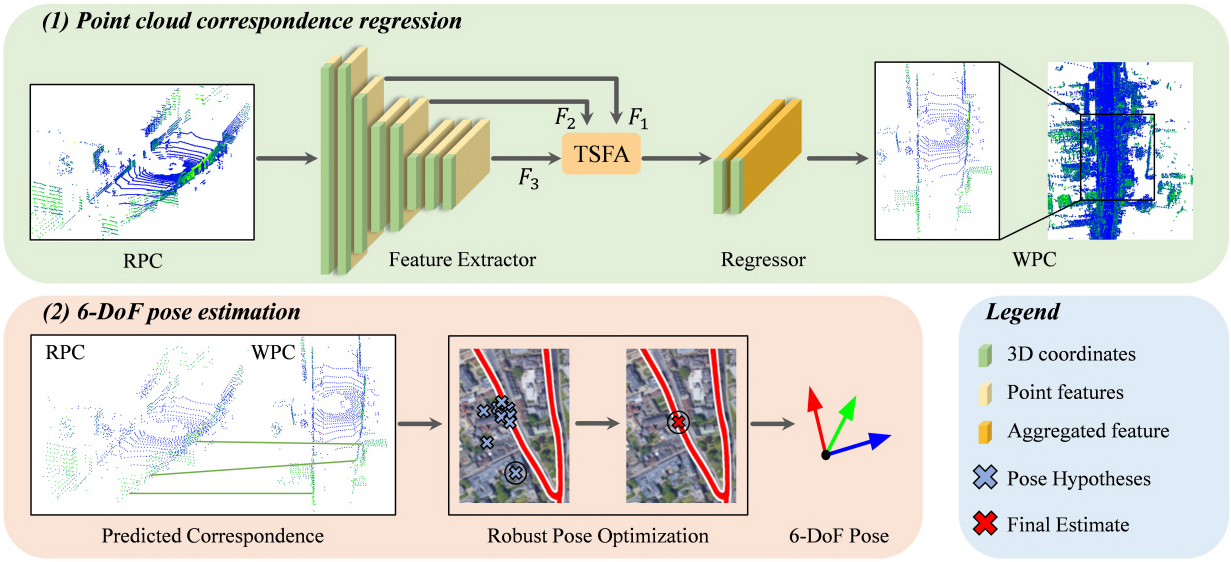

Figure 2. Overview of the proposed framework including point cloud correspondence regression (top row) and 6-DoF pose estimation (bottom row). F_1, F_2, and F_3 are the extracted feature maps with different receptive fields.

rithms, *e.g.*, PGO, NVS, introducing additional computations. Unlike these methods, we decouple the pose estimation process to (1) regression point cloud correspondence and (2) pose estimate via the correspondence using RANSAC. Through step (1), scene geometry can be effectively encoded without additional computations.

Recently, LiDAR-based APR methods, *e.g.*, PointLoc [46], PoseSOE [51], and PosePN++ [51], have shown impressive performance on large-scale outdoor datasets because the point cloud is robust to illumination changes and rich in geometric information. However, effectively encoding scene geometry remains quite challenging.

3. Method

Recent APR methods have achieved impressive results in localization. However, they do not effectively encode the scene geometry and measure the data quality, leading to the accuracy still having room to improve. In this paper, we propose SGLoc and a Pose Quality Evaluation Enhancement (PQEE) method to address these problems. Sec. 3.1 elaborates SGLoc, which can effectively capture the scene geometry. Then, a PQEE method (Sec. 3.2) is proposed to measure and correct the pose errors in the data.

3.1. SGLoc

We now introduce the proposed SGLoc, as shown in Fig. 2, which can be divided into (1) point cloud correspondence regression: convert the Raw LiDAR Point Cloud (RPC) to its corresponding Point Cloud in World coordinate frame (WPC) and (2) 6-DoF pose estimation: estimate pose via matching RPC and WPC. In this work, we design a sparse-convolution-based FCN [15,31] to implement SGLoc, which contains a feature extractor, a Tri-Scale Spatial Feature Aggregation (TSFA) module, and a regressor. The feature extractor and regressor generate feature maps with different receptive fields and achieve point cloud correspondence via regression, respectively. The TSFA module and inter-geometric consistency constraint (IGCC) loss are proposed to effectively capture scene geometry.

Framework. Here we formulate the framework of SGLoc. Given the query point cloud $P_t \in \mathbb{R}^{N\times 3}$, we aim to estimate its global 6-DoF pose $\mathbf{p}$. Each pose $\mathbf{p} = [\mathbf{x}, \mathbf{q}]$ is represented by a position vector $\mathbf{x} \in \mathbb{R}^3$ and an orientation vector $\mathbf{q} \in \mathbb{R}^r$ (*e.g.*, a 4D unit quaternion or a 3D Euler angle). Hence, we first define the query point cloud pose $\mathbf{p}$ as the transformation that maps 3D points in LiDAR coordinate frame, denoted as $\mathbf{l}$, to 3D points in world coordinate frame, denoted as $\mathbf{w}$, i.e.

$$w_i = Tl_i, \tag{1}$$

where i denotes the point index in the query point cloud; T is a 4×4 matrix representation of the pose $\mathbf{p}$.

Then, we denote the complete set of scene coordinates with a global pose for the query point cloud as $\mathcal{Y}$, i.e., $y_i \in \mathcal{Y}$. As shown in Fig. 2, (1) for point cloud correspondence regression, we utilize a neural network to learn a function $\mathscr{F}$: $\mathcal{Y} = \mathscr{F}(P_t)$. (2) Regarding 6-DoF pose estimation during inference, we utilize the RANSAC algorithm to select M group from the predicted correspondence $\mathcal{Y}$, and optimize the energy function as follows:

$$T' = \arg\min_{T} \sum_{i}^{|M|} \|Tl_i - w_i\|_2, \tag{2}$$

where M is set to 3; i denotes the point index; T' is the estimated transformation matrix. Then, T' can be used to calculate the above energy function of each correspondence

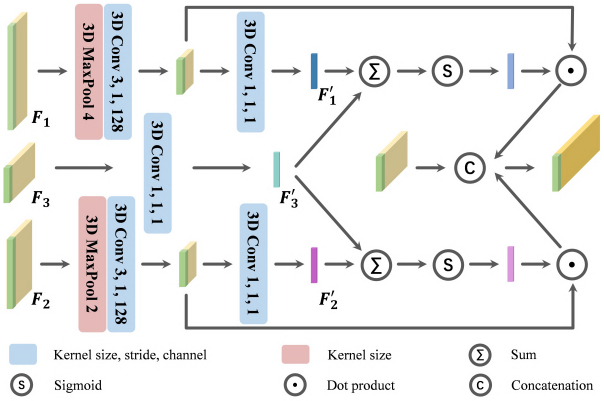

Figure 3. Tri-Scale Spatial Feature Aggregation (TSFA) Module.

in $\mathcal{Y}$ to filter outliers. The above process will be repeated until convergence. Finally, the estimated T' can easily be converted into a vectorized pose.

It is worth noting that during training, SGLoc only requires the input point cloud and the pose, which allows for effective scene geometry encoding by minimizing correspondence distance without any additional computations.

Tri-scale spatial feature aggregation. As described above, SGLoc estimates the pose via the regressed correspondence $\mathcal{Y}$. Therefore, the spatial details in network features are critical to localization accuracy. To effectively capture the spatial details at shallow layers, inspired by the attention mechanism [19, 29, 53], we introduce a Tri-Scale Spatial Feature Aggregation (TSFA) module.

As shown in Fig. 2, the TSFA module takes the low-level features F_1 and F_2 and high-level feature F_3 as the input. We elaborate the architecture of TSFA module in Fig. 3. For convenience, we refer $F'_1 = \mathcal{H}(\mathcal{D}(\mathcal{H}(F_1))$, $F'_2 = \mathcal{H}(\mathcal{D}(\mathcal{H}(F_2))$ and $F'_3 = \mathcal{H}(F_3))$, where $\mathcal{D}$ indicates the down-sampling and $\mathcal{H}$ denotes the convolution followed by a batch normalization and a ReLU activation function. We first squeeze the channels of the features (F_1, F_2, F_3) to 1. Then, we conduct the addition operation and sigmoid activation to obtain the spatial attention mask, which can adaptively enhance spatial details. Finally, we apply dot product and concatenation to achieve a powerful feature with rich spatial details and structure information. The output can be expressed as:

$$F_o = [\sigma(F'_1 + F'_3) \odot F_1, \sigma(F'_2 + F'_3) \odot F_2, F_3], \quad (3)$$

where σ is the sigmoid function; $\odot$ is dot product; $[\cdot]$ denotes the concatenation.

Inter-geometric consistency constraint. Training the network with $l1$ loss ($\mathcal{L}_{L1}$) can effectively minimize the distance between the predicted and the ground truth scene coordinates. However, the above constraint is a node-wise loss, which supervises each correspondence individually. This is unfavorable for learning scene geometry. Inspired by spatial compatibility [1, 13], which assumes that two correspondences have a higher score if the difference of spatial distance between them, we propose an Inter-Geometric Consistency Constraint (IGCC) loss to well learn scene geometry:

$$\mathcal{L}_{IGCC} = \frac{1}{|\mathcal{Y}|^2} \sum_{i,j} \|d_{i,j} - d^*_{i,j}\|_1, \quad (4)$$

where $d^*_{i,j} = \|y^*_i - y^*_j\|_1$ is the ground truth inter-geometric consistency value. This constraint supervises the pairwise distance between the correspondences, serving as a complement to the node-wise supervision.

Loss function. During training, the point cloud with global poses predicted by the network $\mathscr{F}$ is optimized by $\mathcal{L}_{L1}$ and $\mathcal{L}_{IGCC}$. For $\mathcal{L}_{L1}$, we minimize the average plain Euclidean distance between the predicted scene coordinates y_i, and the ground truth of scene coordinates y^*_i:

$$\mathcal{L}_{L1} = \frac{1}{|\mathcal{Y}|} \sum_{y_i \in \mathcal{Y}} \|y_i - y^*_i\|_1. \quad (5)$$

The aforementioned $\mathcal{L}_{IGCC}$ works with $\mathcal{L}_{L1}$ to come up with the final loss:

$$\mathcal{L} = \mathcal{L}_{L1} + \lambda \mathcal{L}_{IGCC}, \quad (6)$$

where λ is a hyper-parameter to balance the two constraints.

Algorithm 1 Pseudocode for PQEE method

Input: Raw point cloud set P_r and pose set T_r; Standard point cloud set P_s and pose set T_s
Output: Pose error E'; Quality enhanced pose set T'_r
Initialization: Build submaps $M_r = \{M_{r_1}, \ldots, M_{r_n}\}$ by P_r and T_r; Build submaps $M_s = \{M_{s_1}, \ldots, M_{s_m}\}$ by P_s and T_s; $E \leftarrow 0$; $N \leftarrow 0$

```
for all M_{r_i} ∈ M_r do
    Search M_{r_i}'s nearest M_{s_j}
    Register to obtain correspondences C and T_ij
    T'_{r_i} ← T_ij T_{s_i}
    for all C_k ∈ C do
        Calculate the Euclidean distance d_k
        E ← E + d_k
        N ← N + 1
return E/N, T'_r
```

3.2. Pose Quality Evaluation and Enhancement

For localization, a high-quality dataset should precisely provide the same position values in the same position on different days. However, existing large-scale outdoor datasets do not satisfy this requirement since various pose errors in the data caused by GPS/INS measuring errors. We empirically find these inaccuracies significantly degrade localization accuracy. Therefore, we propose a Pose Quality Evaluation and Enhancement (PQEE) method to measure and correct pose errors in the data, see Alg. 1.

We first build submaps from point clouds and poses provided by datasets. Then, for each submap of the raw point

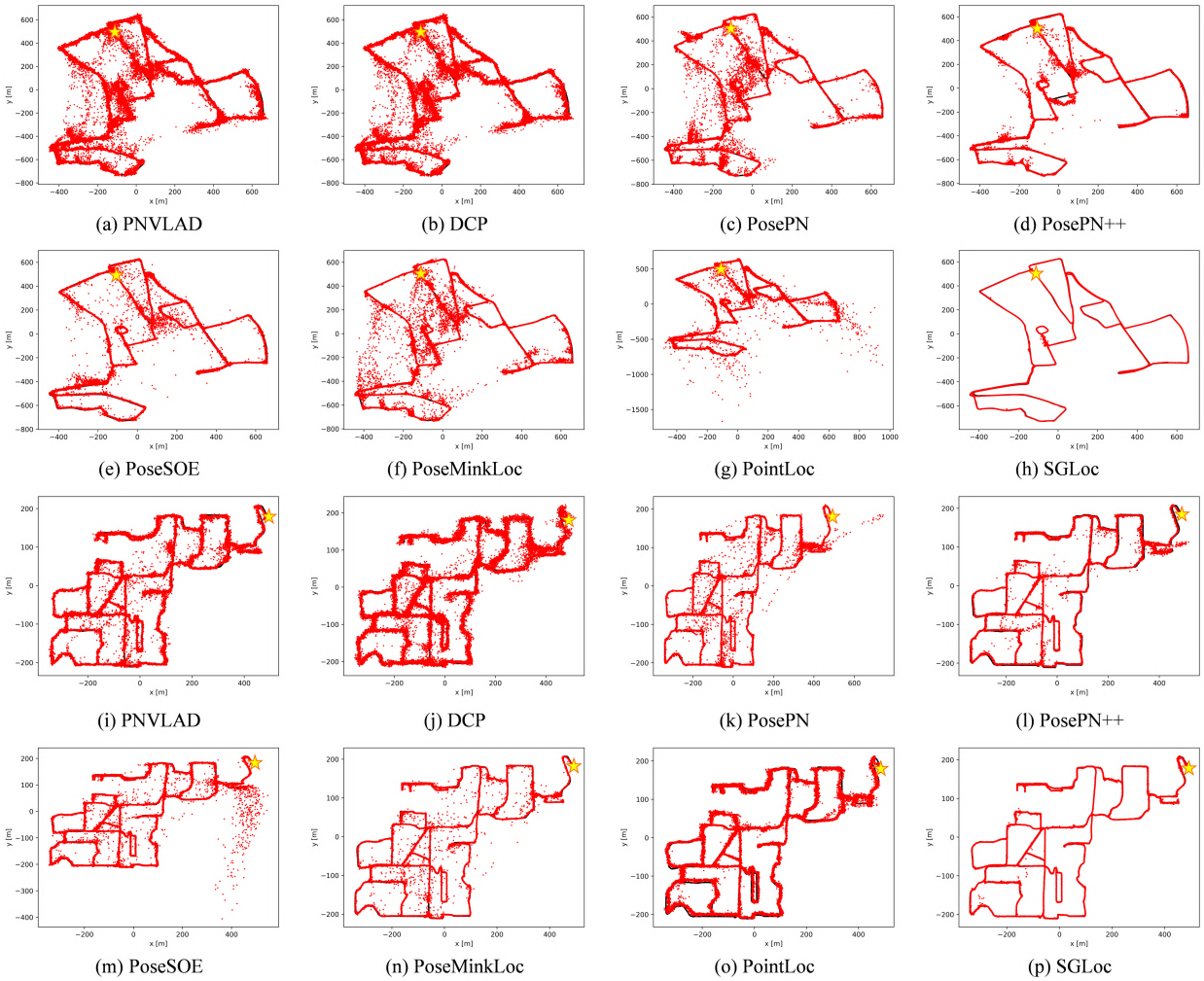

Figure 4. Trajectories of the baselines and the proposed method on the Oxford (top) and NCLT (bottom) datasets. The ground truth and predictions are shown in black and red, respectively. The star indicates the starting position.

cloud, we search for its nearest submap in the standard point cloud, which is a pre-specified day of data, using the provided pose. The submaps are then registered to obtain the correspondence and transformation matrix, which can be used to evaluate and enhance the quality of the pose.

Submaps building. The trajectory is divided into segments based on the prespecified distance to generate the sub-point cloud set. Then, we build the local submap according to Eq. (1), where the position of this submap is defined as the average position of the contained point cloud.

Registration. For each submap M_{r_i}, we search for its nearest submap M_{s_j} based on position. Then, the ICP algorithm [3] is used to align the pair of submaps to obtain correspondence $\mathcal{C}$ and relative transformation matrix T_{ij}.

Pose evaluation and refinement. We calculate the average Euclidean distance between the correspondences of all submaps for pose quality evaluation. For pose refinement, the T_{ij} is used to transform the pose of all point clouds in the M_{ri} to correct pose errors in the data. The algorithm results are convincing when the transformed pose error is below the pre-specified threshold.

4. Experiments

4.1. Settings

Benchmark datasets. We conduct experiments on two large-scale benchmark datasets. Oxford Radar RobotCar [2] (Oxford) is an urban scene localization dataset that contains over 32 repetitions traversals of a center Oxford route (about 10km, 200hm^2). The dataset contains different weather, traffic, and lighting conditions. The point cloud is collected by dual Velodyne HDL-32E LiDAR. Ground truth pose is generated by the interpolations of INS. We use the data of 11-14-02-26, 14-12-05-52, 14-14-48-55, and 18-15-20-12 as the training set. The data of 15-13-06-37, 17-13-26-39, 17-14-03-00, and 18-14-14-42 are used as the test data.

NCLT [34] dataset is collected by sensors on a Segway

Methods	PNVLAD	DCP	PosePN	PosePN++	PoseSOE	PoseMinkLoc	PointLoc	**SGLoc**
Oxford dataset								
15-13-06-37	18.14m, 3.28°	16.04m, 4.54°	14.32m, 3.06°	9.59m, 1.92°	7.59m, 1.94°	11.20m, 2.62°	12.42m, 2.26°	**3.01m**, **1.91°**
17-13-26-39	24.57m, 3.08°	16.22m, 3.56°	16.97m, 2.49°	10.66m, **1.92°**	10.39m, 2.08°	14.24m, 2.42°	13.14m, 2.50°	**4.07m**, 2.07°
17-14-03-00	19.93m, 3.13°	14.87m, 3.45°	13.48m, 2.60°	9.01m, **1.51°**	9.21m, 2.12°	12.35m, 2.46°	12.91m, 1.92°	**3.37m**, 1.89°
18-14-14-42	15.59m, 2.63°	12.97m, 3.99°	9.14m, 1.78°	8.44m, 1.71°	7.27m, 1.87°	10.06m, 2.15°	11.31m, 1.98°	**2.12m**, **1.66°**
Average	19.56m, 3.03°	15.03m, 3.89°	13.48m, 2.48°	9.43m, **1.77°**	8.62m, 2.00°	11.96m, 2.41°	12.45m, 2.17°	**3.14m**, 1.88°
Quality-enhanced Oxford dataset								
15-13-06-37	10.90m, 2.49°	10.61m, 2.56°	9.47m, 2.80°	4.54m, 1.83°	4.17m, 1.76°	6.77m, 1.84°	10.75m, 2.36°	**1.79m**, **1.67°**
17-13-26-39	14.60m, 2.46°	11.44m, 2.14°	12.98m, 2.35°	6.44m, 1.78°	6.16m, 1.81°	8.84m, 1.84°	11.07m, 2.21°	**1.81m**, **1.76°**
17-14-03-00	11.28m, 2.21°	10.90m, 2.01°	8.64m, 2.19°	4.89m, **1.55°**	5.42m, 1.87°	8.08m, 1.69°	11.53m, 1.92°	**1.33m**, 1.59°
18-14-14-42	9.00m, 1.90°	9.51m, 2.08°	6.26m, 1.64°	4.64m, 1.61°	4.16m, 1.70°	6.56m, 2.06°	9.82m, 2.07°	**1.19m**, **1.39°**
Average	11.45m, 2.27°	10.62m, 2.20°	9.34m, 2.25°	5.13m, 1.69°	4.98m, 1.79°	7.56m, 1.86°	10.79m, 2.14°	**1.53m**, **1.60°**

Table 1. Position error (m) and orientation error (°) for various methods on the Oxford and quality-enhanced Oxford datasets.

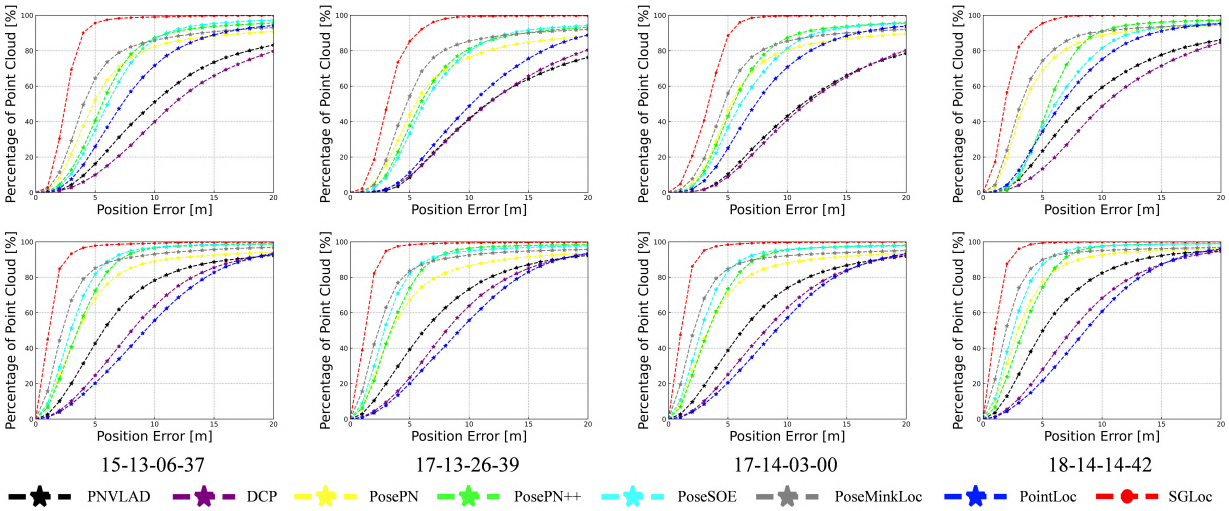

Figure 5. Cumulative distributions of the position errors (m) on the Oxford and quality-enhanced Oxford dataset. The x-axis is the position error, and the y-axis is the percentage of point clouds with errors less than the value.

robotic platform on the University of Michigan's North Campus. The dataset contains 27 traversals, where each traversal is nearly 5.5km and covers 45hm^2. The point cloud is gathered by a Velodyne HDL-32E LiDAR. The 6-DoF ground truth pose is obtained by SLAM. The data of 2012-01-22, 2012-02-02, 2012-02-18, and 2012-05-11 are treated as the training set, and the data of 2012-02-12, 2012-02-19, 2012-03-31, and 2012-05-26 are used as the test set. More details about datasets can be found in the supplementary.

Implementation details. The proposed SGLoc is implemented with PyTorch [35] and Minkowski Engine [15]. We run our code on a PC equipped with an Intel (R) Xeon (R) Silver 4214R CPU, 64GB of RAM, and a single NVIDIA RTX 3090 GPU. During training, we employ an Adam optimizer [26] with an initial learning rate of 0.001. The weight λ is set to 1 for all datasets. On the Oxford dataset, we use the point cloud from the left LiDAR and set the voxel size to 0.2m. On the NCLT dataset, the voxel size is 0.25m. The TSFA module uses feature maps from the 3rd, 5th, and 8th convolution blocks with 128, 256, and 512 dimensions, respectively. Each convolution block consists of two convolution layers and a residual connection.

Baselines. To validate the performance of SGLoc, we compare it with several state-of-the-art learning-based LiDAR localization approaches. PointNetVLAD (PNVLAD) [44] is a large-scale point cloud retrieval-based method, and DCP [47] is a point cloud registration approach which employs PointNet [36] and DGCNN [48] as the embedding network. PNVLAD and DCP use the same configuration as PointLoc [46]. PosePN [51], PosePN++ [51], PoseSOE [51], PoseMinkLoc [51], and PointLoc are LiDAR-based localization framework that uses a single-frame point cloud for absolute pose regression.

Results of pose quality evaluation and enhancement. On the Oxford and NCLT datasets, the data of 14-14-48-55 and 2012-02-18 are selected as the standard, respectively. We set the distance of the trajectory segments to build sub-maps to 20m and the voxel size to 0.1m to reduce the duplicate points. Registration is considered successful when the root-mean-square error of correspondence is below 1m. The evaluated pose error of Oxford and NCLT datasets are 3.24m and 0.25m, respectively. Obviously, the ground truth poses in the data of NCLT is accurate. However, the pose of the Oxford dataset has various errors. Therefore, we perform quality enhancement on the Oxford dataset and obtain the enhanced pose with errors of 0.91m.

Methods	PNVLAD	DCP	PosePN	PosePN++	PoseSOE	PoseMinkLoc	PointLoc	**SGLoc**
2012-02-12	7.75m, 6.49°	9.84m, 6.84°	9.45m, 7.47°	4.97m, 3.75°	13.09m, 8.05°	6.24m, 5.03°	7.23m, 4.88°	**1.20m**, **3.08°**
2012-02-19	7.47m, 5.49°	8.27m, 5.16°	6.15m, 5.05°	3.68m, **2.65°**	6.16m, 4.51°	4.87m, 3.94°	6.31m, 3.89°	**1.20m**, 3.05°
2012-03-31	6.98m, 5.67°	8.94m, 5.96°	5.79m, 5.28°	4.35m, 3.38°	5.24m, 4.56°	4.23m, 4.03°	6.71m, 4.32°	**1.12m**, **3.28°**
2012-05-26	14.34m, 7.93°	15.62m, 7.99°	13.47m, 7.77°	9.59m, **4.49°**	12.60m, 7.67°	10.32m, 6.52°	10.02m, 5.32°	**3.81m**, 4.74°
Average	9.14m, 6.40°	10.67m, 6.49°	8.72m, 6.39°	5.65m, 3.57°	9.27m, 6.20°	6.42m, 4.88°	7.57m, 4.60°	**1.83m**, **3.54°**

Table 2. Position error (m) and orientation error (°) for various methods on the NCLT dataset.

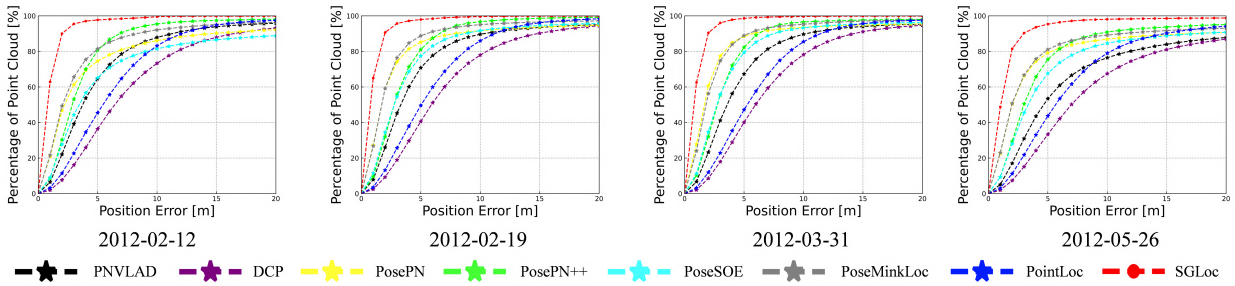

Figure 6. Cumulative distributions of the position errors (m) on the NCLT dataset. The x-axis is the position error, and the y-axis is the percentage of point clouds with errors less than the value.

4.2. Localization Results

Localization on the Oxford dataset. We first evaluate the proposed SGLoc on the Oxford dataset, as shown in Tab. 1. We report the mean position and orientation errors over the full test trajectories. The proposed framework achieves a mean error of 3.14m/1.88°, outperforming all competitors. Specifically, compared to PoseSOE, it improves by 63.6% and 6% on position and orientation, respectively.

We further evaluate the proposed SGLoc on the quality-enhanced Oxford dataset, as shown in Tab. 2. Clearly, the mean position and orientation errors of all methods are significantly degraded compared to Tab. 1, illustrating the effectiveness and generalizability of the proposed PQEE method. SGLoc achieves a mean error of 1.53m/1.60°, surpassing other methods by a large margin. Specifically, it improves PoseSOE by 69.3%/10.6%. These results demonstrate that the proposed SGLoc can perform localization well in large-scale outdoor scenes.

The first two rows in Fig. 4 illustrate the predicted trajectories on 17-14-03-00. The trajectory of SGLoc closely overlaps with the ground truth, demonstrating its accuracy. Fig. 5 shows the cumulative distributions of position errors on different trajectories. SGLoc achieves the desired performance, indicating that it effectively captures the scene geometry, leading to improved accuracy.

Localization on the NCLT dataset. Then, we evaluate the proposed SGLoc on the NCLT dataset. Tab. 2 summarizes the results of all methods with mean position and orientation errors. Apparently, SGLoc significantly outperforms the existing methods with a mean error of 1.83m/3.54°. Our approach improves by 67.6% on position compared to PosePN++. It should be explained that there are more outliers on 2012-05-26 due to the various differences between the test and training trajectories.

The last two rows of Fig. 4 show the predicted trajectory of 2012-03-31. Our prediction is closer to the ground truth and smoother than the competitors. The cumulative distributions of position errors are shown in Fig. 6, demonstrating the promising performance of the proposed SGLoc. It means that SGLoc can capture the scene geometry more efficiently compared to existing LiDAR-based localization methods.

	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42
SGLoc w/o PGO	1.79m, 1.67°	1.81m, 1.75°	1.33m, 1.59°	1.19m, 1.39°
SGLoc w/ PGO	1.58m, 1.10°	1.56m, 1.16°	1.10m, 1.18°	**0.99m**, 1.04°
	2012-02-12	2012-02-19	2012-03-31	2012-05-26
SGLoc w/o PGO	1.20m, 3.08°	1.20m, 3.05°	1.12m, 3.28°	3.81m, 4.74°
SGLoc w/ PGO	**0.88m**, 2.35°	**0.85m**, 2.06°	**0.79m**, 2.34°	3.25m, 3.52°

Table 3. Localization results of SGLoc with PGO on the quality-enhanced Oxford and NCLT datasets.

4.3. Localization Accuracy at Sub-meter Level

Similar to MapNet [8], we utilize PGO as post-processing to further improve localization results. As shown in Tab. 3, our method has further improved in accuracy with PGO. Specifically, the SGLoc achieves accuracy at the sub-meter level on 18-14-14-42 of the quality-enhanced Oxford dataset. Moreover, in all scenes of the NCLT dataset (except 2012-05-26), the mean error of SGLoc is 0.84m/2.25°. To our knowledge, SGLoc is the first regression-based method to reduce the error to the level of the sub-meter on some trajectories, which bridges the gaps in practical applications. For more results, please refer to the supplementary.

4.4. Ablation Study

Ablation on PQEE. As shown in Tab. 4, the proposed method outperforms the vanilla model without proposed modules by 43.5%/14.9%. Moreover, the performance of PQEE is significantly enhanced with or without IGCC or TSFA. For instance, PQEE considerably surpasses that without PQEE under IGCC and TSFA, achieving a mean error of 1.53m/1.60° vs. 3.14m/1.88°.

	IGCC	TSFA	Oxford w/o PQEE	Oxford w/ PQEE	NCLT
1			3.95m/2.89°	2.23m/2.46°	2.73m/5.19°
2	✓		3.46m/2.67°	2.12m/2.32°	2.62m/5.11°
3		✓	3.23m/2.10°	1.68m/1.67°	1.98m/**3.51**°
4	✓	✓	**3.14m/1.88**°	**1.53m/1.60**°	**1.83m**/3.54°

Table 4. Ablation study on the Oxford and NCLT datasets.

Ablation on IGCC. We further conduct ablation experiments to demonstrate the importance of IGCC loss. In Tab. 4, the comparison between Row 1 and Row 2 shows the module notably improves accuracy. On the Oxford dataset, compared to the vanilla model, IGCC obtains a 12.4%/7.6% improvement. On the NCLT dataset, IGCC also leads to improvements. Furthermore, IGCC improves localization accuracy compared to the method with TSFA only. This demonstrates the proposed design is helpful for capturing the scene geometry.

Ablation on TSFA. The TSFA results are reported in Row 3 of Tab. 4. On the Oxford and NCLT dataset, compared to the results in Row 1, it yields an average 23.5%/30.6% improvement on position and orientation. This significant improvement verifies that the proposed TSFA module can further improve the encoding of scene geometry.

4.5. Runtime

For the Oxford and NCLT datasets, the LiDAR scan rate is 20Hz and 10Hz, respectively. Therefore, the real-time performance here means that the running time of each scanning data is less than 50ms and 100ms, respectively. Tab. 5 shows the running time on the Oxford and NCLT dataset. On the Oxford dataset, SGLoc takes a running time of 38ms per frame, with the correspondence regression stage consuming 25ms and the pose estimation stage consuming 13ms. On the NCLT dataset, the running times are 75ms, 41ms, and 34ms, respectively. Note that the running time of PNVLAD, DCP, PosePN, and PoseMinkLoc are less than 10ms due to the simplicity of their networks. Therefore, Our SGLoc can achieve real-time localization and is very competitive with these methods, but the performance of SGLoc is much better.

5. Discussion

LiDAR localization in large-scale outdoor scenes is a critical component of autonomous driving. However, existing APR methods still face challenges in effectively encoding scene geometry and dealing with unsatisfactory data quality, resulting in suboptimal accuracy. To overcome these challenges, we propose SGLoc and PQEE, which significantly improves accuracy. Compared to previous work, SGLoc stands out due to the following differences.

Difference between SGLoc and DSAC++. Camera coordinate regression (DSAC++ [5]) predicts 2D-3D correspondences, to our knowledge, SGLoc is the first work to regress 3D-3D point correspondences. Though both DSAC++ and our SGLoc use regression, we address different problems.

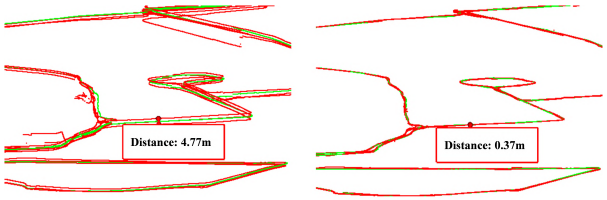

Figure 7. Ground truth poses in the data on the Oxford (left) and quality-enhanced Oxford (right) dataset. The green trajectory indicates the prespecified standard data.

Methods	PNVLAD	DCP	PosePN	PosePN+	PoseSOE	PoseMinkLoc	PointLoc	**SGLoc**
Oxford	6ms	3ms	2ms	111ms	244ms	8ms	625ms	38ms
NCLT	6ms	3ms	2ms	108ms	230ms	8ms	614ms	75ms

Table 5. Runtime (ms) of different methods on the Oxford and NCLT datasets.

We summarize the differences. (1) SGLoc learns rigid transformations instead of affine transformations that require the intrinsics of the query camera; (2) Our methods' ground truth correspondences generation requires only input point clouds and poses, which can be deployed efficiently in online training; (3) SGLoc can be applied to large-scale outdoor scenes covering about $200\mathrm{hm}^2$.

Difference between PQEE and point cloud registration. We empirically find existing large-scale outdoor datasets cannot precisely provide the same position values (bad quality) in the same position on different days, as shown in Fig. 7. More importantly, we find this degraded data quality can significantly decrease the localization accuracy. We hope this new finding can inspire researchers to work on this new source of performance degradation. Based on this new finding, we propose PQEE, which aligns and fuses standard/raw data in LiDAR localization. It uses point cloud registration, but the purpose differs from conventional point cloud registration.

6. Conclusion

We propose a novel regression-based framework, SGLoc, for LiDAR localization. SGLoc is the first work to decouple LiDAR localization into point cloud correspondences regression and pose estimation via predicted correspondences. The core component of SGLoc is the first step, which effectively encodes scene geometry, leading to significant performance improvement. To achieve high accuracy in this step, we propose the TSFA module and IGCC loss to improve the encoding of scene geometry. Moreover, to increase the data quality and localization accuracy, the PQEE method is designed to measure and correct the ground truth pose errors in the localization data. Extensive experimental results verify the great effectiveness of our method.

Acknowledgements This work was partially supported by the open fund of PDL (2022-KJWPDL-12, WDZC20215250113), by the FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform (No.3502ZCQXT2021003).

References

[1] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, pages 15859–15869, 2021. 4

[2] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *ICRA*, pages 6433–6438, 2020. 1, 2, 5

[3] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606, 1992. 5

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac - differentiable ransac for camera localization. In *CVPR*, pages 6684–6692, 2017. 1, 2

[5] Eric Brachmann and Carsten Rother. Learning less is more - 6d camera localization via 3d surface regression. In *CVPR*, pages 4654–4662, 2018. 2, 8

[6] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, pages 7525–7534, 2019. 2

[7] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE TPAMI*, 44:5847–5865, 2021. 2

[8] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, pages 2616–2625, 2018. 1, 2, 7

[9] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet: Deep loop closure detection and point cloud registration for lidar slam. *IEEE TR*, 2022. 2

[10] Shuai Chen, Xinghui Li, Zirui Wang, and Victor Prisacariu. Dfnet: Enhance absolute pose regression with direct feature matching. In *ECCV*, 2022. 1, 2

[11] Shuai Chen, Zirui Wang, and Victor Prisacariu. Direct-posenet: Absolute pose regression with photometric consistency. In *3DV*, pages 1175–1185, 2021. 1, 2

[12] Xieyuanli Chen, Thomas Läbe, Lorenzo Nardi, Jens Behley, and Cyrill Stachniss. Learning an overlap-based observation model for 3d lidar localization. In *IROS*, pages 4602–4608, 2020. 2

[13] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *CVPR*, pages 13221–13231, 2022. 4

[14] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, pages 2514–2523, 2020. 2

[15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 3, 6

[16] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, pages 8958–8966, 2019. 2

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 15:381–395, 1981. 2

[18] Zhixing Hou, Yan Yan, Chengzhong Xu, and Hui Kong. Hitpr: Hierarchical transformer for place recognition in point cloud. In *ICRA*, 2022. 1, 2

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 4

[20] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *CVPR*, pages 4267–4276, 2021. 2

[21] Zhaoyang Huang, Yan Xu, Jianping Shi, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *ICCV*, pages 2791–2800, 2019. 1

[22] Zhaoyang Huang, Han Zhou, Yijin Li, Bangbang Yang, Yan Xu, Xiaowei Zhou, Hujun Bao, Guofeng Zhang, and Hongsheng Li. Vs-net: Voting with segmentation for visual localization. In *CVPR*, pages 6101–6111, 2021. 2

[23] Le Hui, Hang Yang, Mingmei Cheng, Jin Xie, and Jian Yang. Pyramid point cloud transformer for large-scale place recognition. In *ICCV*, pages 6098–6107, 2021. 1

[24] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, pages 5974–5983, 2017. 1, 2

[25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, page 2938–2946, 2015. 1, 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[27] Jacek Komorowski. Minkloc3d: Point cloud based large-scale place recognition. In *WACV*, pages 1790–1799, 2021. 1, 2

[28] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, pages 11983–11992, 2020. 2

[29] Dongfang Liu, Yiming Cui, Mousas Christos Yan, Liqi, Baijian Yang, and Yingjie Chen. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In *AAAI*, pages 6101–6109, 2021. 4

[30] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, pages 2831–2840, 2019. 2

[31] Jonathan Long, Evan Shelhamer, and Trevoro Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3075–3084, 2015. 3

[32] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *ICCV*, pages 16014–16023, 2021. 2

[33] Mohamed Adel Musallam, Vincent Gaudillière, Miguel Ortiz del Castillo, Kassem Al Ismaeil, and Djamila Aouada. Leveraging equivariant features for absolute pose regression. In *CVPR*, pages 6876–6886, 2021. 2

[34] Carlevaris-Bianco Nicholas, K. Ushani Arash, and M. Eustice Ryan. University of michigan north campus long-term vision and lidar dataset. *Int. J. of Rob. Res.*, 35:545–565, 2015. 1, 2, 5

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[36] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 6

[37] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, pages 11143–11152, 2022. 2

[38] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *CVPR*, pages 3302–3312, 2019. 1, 2

[39] Yoli Shavit and Ron Ferens. Introduction to camera pose estimation with deep learning. *arXiv preprint arXiv:1907.05272*, 2019. 1, 2

[40] Yoli Shavit, Ron Ferens, and Yosi Keller. Learning multi-scene absolute pose regression with transformers. In *CVPR*, pages 2733–2742, 2021. 1, 2

[41] Yoli Shavit and Yosi Keller. Camera pose auto-encoders for improving pose regression. In *ECCV*, 2022. 2

[42] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *CVPR*, pages 2930–2937, 2013. 2

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2

[44] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, pages 4470–4479, 2018. 2, 6

[45] Bing Wang, Chaohao Chen, Chrisxiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, pages 10393–10401, 2020. 1, 2

[46] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. Pointloc: Deep pose regressor for lidar point cloud localization. *IEEE Sensors*, 22:959–968, 2022. 3, 6

[47] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *CVPR*, pages 3523–3532, 2019. 2, 6

[48] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM TOG*, 38, 2019. 6

[49] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stilla. Soe-net: A self-attention and orientation encoding network for point cloud based place recognition. In *CVPR*, pages 11348–11357, 2021. 1, 2

[50] Huan Yin, Yue Wang, Xiaqing Ding, Li Tang, Shoudong Huang, and Rong Xiong. 3d lidar-based global localization using siamese neural network. *IEEE TITS*, 21(4):1380–1392, 2019. 2

[51] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. Lidar-based localization using universal encoding and memory-aware regression. *PR*, 128:108915, 2022. 1, 3, 6

[52] Shangshu Yu, Cheng Wang, Zenglei Yu, Xin Li, Ming Cheng, and Yu Zang. Deep regression for lidar-based localization in dense urban areas. *ISPRS J. Photogramm Remote Sens.*, 172:240–252, 2021. 2

[53] Zhimin Yuan, Chenglu Wen, Ming Cheng, Yanfei Su, Weiquan Liu, Shangshu Yu, and Cheng Wang. Category-level adversaries for outdoor lidar point clouds cross-domain semantic segmentation. *IEEE TGRS*, 2022. 4

[54] Lei Zhou, Zixin Luo, Tianwei Shen, Jiahui Zhang, Mingmin Zhen, Yao Yao, Tian Fang, and Long Quan. Kfnet: Learning temporal camera relocalization using kalman filtering. In *CVPR*, pages 4919–4928, 2020. 2