Unleashing the Power of Data Generation in One-Pass Outdoor LiDAR Localization

Yidong Chen
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
Key Laboratory of Multimedia Trusted Perception and Efficient Computing
Ministry of Education of China, Xiamen University
Xiamen, China
chenyidong@stu.xmu.edu.cn

Qi Li
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
Key Laboratory of Multimedia Trusted Perception and Efficient Computing
Ministry of Education of China, Xiamen University
Xiamen, China
qbbov0@stu.xmu.edu.cn

Yuyang Yang
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
Key Laboratory of Multimedia Trusted Perception and Efficient Computing
Ministry of Education of China, Xiamen University
Xiamen, China
yuyangyang@stu.xmu.edu.cn

Wen Li
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
liwen777@stu.xmu.edu.cn

Sheng Ao
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
aosh@xmu.edu.cn

Cheng Wang*
Fujian Key Laboratory of Sensing and Computing for Smart Cities
Xiamen University
Xiamen, China
cwang@xmu.edu.cn

Abstract

Point cloud regression localization technology has a wide range of applications in the multimedia field. For example, in virtual reality and augmented reality, accurate point cloud localization can significantly enhance the user experience. Recently, point cloud pose regression algorithms based on APR (Absolute Pose Regression) and SCR (Scene Coordinate Regression) have achieved near sub-meter accuracy, requiring multiple repetitive trajectories for training. The key to their success lies in the diversity of viewpoints, temporal changes, and trajectories,**which is resource-consuming**. However, due to the errors in GPS/INS, the coupling between trajectories is not ideal, and the stability of re-localization is insufficient. Since LiDAR has covered most of the scene, single-shot localization has the potential to approach or even surpass multi-trajectory localization methods through pose enhancement. Specifically, we present Pose Enhancement Localization (PELoc), which feeds one trajectory, proposing SSDA (Single-shot Data Augmentation) and LTI (LiDAR Trajectories-coupled Interpolation) to simulate different driving poses, and we introduce KP-CL (Key Points Contrastive Learning) through feature perturbation to mitigate the differences in viewpoint/temporal phase transformations in similar scenes across different trajectories. Our algorithm has been tested on the Oxford, QE-Oxford, and NCLT datasets, where single-shot localization accuracy can approach near sub-meter level on QE-Oxford and NCLT. The code will be published in *https://github.com/Eaton2022/PELoc*.

CCS Concepts

• **Computing methodologies** → **Vision for robotics**.

Keywords

Single-shot localization, Single-shot Data Augmentation, LiDAR Trajectories-coupled Interpolation, Key Points Contrastive Learning

ACM Reference Format:
Yidong Chen, Qi Li, Yuyang Yang, Wen Li, Sheng Ao, and Cheng Wang. 2025. Unleashing the Power of Data Generation in One-Pass Outdoor LiDAR Localization. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3754968

1 Introduction

Multi-path effects and terrain occlusions can degrade the performance of GPS/GNSS devices, limiting their ability to provide high-precision localization[10, 18] in all conditions[16, 28]. Recently, point cloud localization[37, 50] based on deep learning has enabled scene recognition when devices fail. Therefore, point cloud[12, 24] localization is a fundamental and hot research topic in fields such as autonomous driving[7, 11, 22, 34] and intelligent navigation[8, 35, 51–53]. Retrieval-based methods[27, 32, 41] require pre-built maps, which demand substantial storage and computational resources for maintaining and searching the map database.

*Corresponding author

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754968

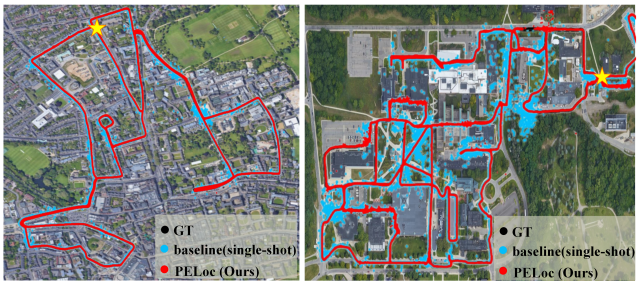


Figure 1: Single-shot localization results of PELoc (ours) and the baseline on the Oxford and NCLT datasets. PELoc achieves significantly more accurate localization results.

In contrast, absolute pose regression[31] methods are map-free, enabling efficient and rapid decision-making for localization.

Popular regression-based localization algorithms[21, 46] heavily rely on multiple repetitive trajectories, which provide rich scene and viewpoint variations. However, trajectory collection is time-consuming and labor-intensive[13, 56]. When the network is fed with only a single trajectory, the model's performance is significantly lower than that of multi-trajectory training. The blue trajectory points in Figure. 1 represents the results of single-shot training, which exhibit a large number of prediction noise points.

Next, we will delve deep into two questions. **Question one**:**Why does single-shot localization fall short in accuracy?** Firstly, the poses between different trajectories inherently have deviations. When using a single trajectory for training, there will inevitably be translation and rotation errors. **Secondly**, the model encounters some reversed or unfamiliar scenes that it has never seen before. **Thirdly**, viewpoint and scene transformations can also cause feature differences in the same scene across different trajectories, which the model finds hard to handle.

Question 2: Why does single-shot localization has the potential to approach even outperform multi-trajectory localization? Some study [20] indicates that the acquisition of ground truth in current methods largely relies on GPS/INS, which is error-prone, as shown in Figure 2 (a). Such ground truth errors also lead to inconsistencies in trajectories, thereby reducing localization accuracy. We conducted this experiment in Figure 2 (d), often necessitating time-consuming trajectory correction. However, even after correcting the poses, there are still a few errors remaining, as shown in Figure 2 (b). This has motivated us to propose LTI for trajectory generation. The trajectories we generate are coupled with each other, as illustrated in Figure 2 (c), resulting in higher relocalization accuracy, a finding that has been verified in our experimental section. Therefore, our trajectories-coupled interpolation has the potential to surpass multi-trajectory localization methods.

Based on the aforementioned analysis, we present PELoc for single-shot localization. The primary limitations of single-shot localization stem from the singularity of trajectory and pose. The antidote lies in: (1) SSDA (single-shot data augmentation) and (2) LTI (LiDAR Trajectories-coupled Interpolation). Additionally, to mitigate the feature differences caused by viewpoint and scene changes, we introduce KP-CL (Key Points Contrastive Learning).

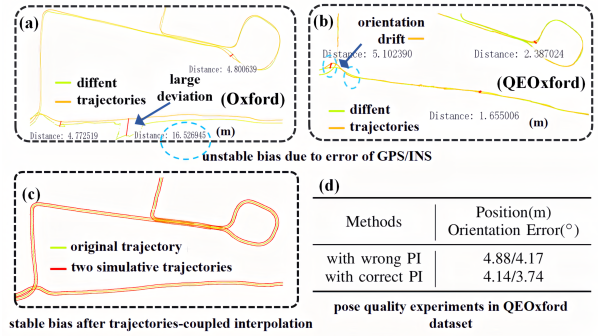


Figure 2: (a) and (b): Due to the errors in GPS/INS devices, the relative stability between different trajectories is not ideal. (c) illustrates the effect of our proposed trajectories-coupled interpolation. (d) presents the experimental results of pose quality. PI denotes pose interpolation.

Our contributions are summarized as follows:

- We conduct a thorough analysis of the key factors and limitations of multi-trajectory point cloud localization, presenting our novel single-shot localization algorithm, PELoc, which is the first work to research map-free single-shot localization as far as we are aware.
- We introduce novel LTI (LiDAR Trajectories-coupled Interpolation), SSDA (Single-shot Data Augmentation), KP-CL (Key Points Contrastive Learning), and a training strategy RFT(Remove 5% Random Consecutive Frames Training) to increase the diversity of data and poses, effectively enhancing the accuracy of single-shot localization.
- We perform experiments on the Oxford, QEOxford, and NCLT datasets, achieving average near sub-meter localization accuracy on the QEOxford (7.15m/4.94° to 1.09m/1.62°) and NCLT datasets (17.94m/10.83° to 1.02m/3.61°), which outperforms the result of the baseline trained with four trajectories in some trajectories. We also compare it with other methods suitable for single-shot localization, thereby demonstrating the superiority of our approach.

2 Related Work

Point cloud localization[23, 30, 38] can be categorized into map-based and regression-based(map-free) methods. Map-based methods treat point cloud localization as a place recognition task[1, 55], following a pipeline of retrieval and registration. Retrieval-based methods rely on pre-built map databases, while registration-based methods can achieve precise localization but at the cost of higher complexity. Regression-based methods can be further divided into relative and absolute pose estimation. Relative pose estimation[17] analyzes consecutive laser scans and is prone to cumulative errors, whereas absolute pose estimation[4, 5] enables independent decision-making. Our single-shot localization is based on absolute pose estimation, which we will focus on introducing next.

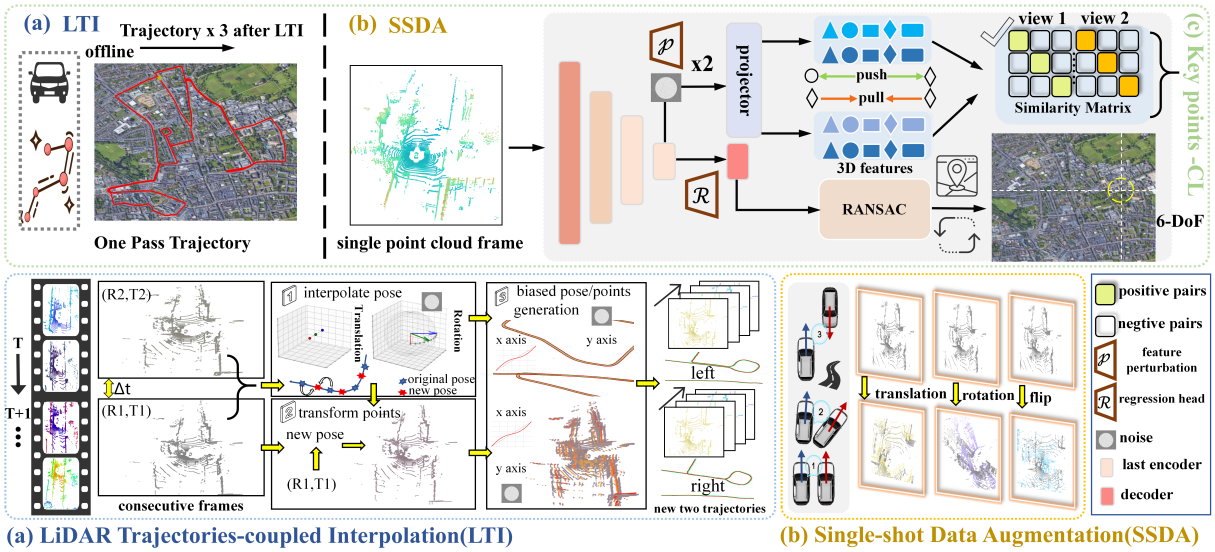


Figure 3: Network architecture of PELoc. PELoc is an algorithm designed for regressing world poses using a single trajectory for long-term localization. The flow of PELoc's network is as follows: LTI generates two new trajectories, making a total of three trajectories for training. After the point clouds undergo SSDA, they are fed into an Encoder-Decoder network. The features extracted are used for both KP-CL and point cloud regression. The final loss is a combination of the losses from KP-CL and point cloud regression. Note: Details of the encoder architecture and SSDA can be found in Figure 4 and in the Experimental Settings section.

2.1 Regression-based Localization

Point cloud regression localization can be divided into APR and SCR methods[20, 48]. APR methods regress the 6-DoF pose directly through a neural network. Classic representative works include Atloc[29], Ms-Transformer[14], PointLoc[33], etc. Recently, Diffloc[19] has treated LiDAR localization as a conditional pose generation problem, utilizing a diffusion model to construct a denoising process for APR. Inspired by the image localization method ACE[3], FlashMix[9] employs a scene-agnostic backbone to extract a feature buffer, significantly reducing training time. SCR methods first regress the 3-DoF world coordinates and then use RANSAC[26] to compute the correspondences between local and world coordinates, resulting in a 6-DoF pose[45]. Representative works include SGLoc and LiSA[40]. SGLoc identified GPS/INS errors in localization datasets, while LiSA incorporated semantic features into the localization process. SCR methods tend to converge faster than APR methods. Our PELoc method is based on SCR. The coupling in our proposed LTI method is better than that of the original multi-trajectories, thus having the potential to surpass the accuracy of multi-trajectory training.

2.2 Single-shot Localization Solutions

To the best of our knowledge, this work is the first to investigate single-shot point cloud map-free regression. We have identified several promising approaches for single-shot localization[42] and compared with them in our ablation studies. (1) Equivariant Features: Point cloud retrieval is typically single-shot. RING[39] derived rotation-translation equivariant point cloud feature representations using Fourier formulas. Kaba et al.[15] learned canonical representations of point cloud rotation data, effectively capturing their equivariance. (2) Point Cloud Interpolation: Due to the insufficient number of samples for a single trajectory, we considered the method of point cloud interpolation. PointINet[25] estimated the scene flow between two adjacent point clouds and proposed a novel point fusion scheme; NeuralPCI[54] further considered the spatiotemporal dimension, implicitly integrating information from multiple frames to handle large nonlinear motions. (3) Contrastive Learning: Contrastive learning has been extensively applied in recommendation systems[44], graph learning[49], and vision tasks[43]. In the 3D detection field, it can effectively identify regions of interest[36], significantly reducing computational load. How to apply contrastive learning in point cloud localization remains a challenge.

We will compare our method with these approaches in the methodology and experimental sections, demonstrating that our method is superior and more suitable for point cloud absolute localization.

3 Proposed Framework: PELoc

3.1 Framework Overview & Motivation

PELoc trains using only one original trajectory. After regressing the world coordinates, it employs RANSAC to eliminate outliers, thereby obtaining a 6-DoF (Degree of Freedom) pose. Recent regression-based LiDAR pose estimation algorithms[20, 40] heavily

rely on the diversity provided by multiple trajectories. However, collecting multiple trajectories is resource-consuming. This has motivated us to explore localization using a single trajectory. To address challenges such as trajectory variations and viewpoint changes, we present PELoc, see Figure. 3. PELoc's LTI and SSDA effectively increase the diversity of training data, while KP-CL's contrastive learning capability further enhances the robustness of the algorithm. Additionally, we propose a new training method, RFT(Remove 5% Random Consecutive Frames for Training), to mitigate the trajectory dependency in single-trajectory training. When using multiple trajectories for localization, data augmentation plays a limited role due to the abundance of data, and some current state-of-the-art methods[20, 40] do not employ it. However, for single-shot localization, data augmentation is critical, prompting us to propose an effective SSDA (Single-shot Data Augmentation). Some current works[57] only consider the rotation and translation of point clouds, but do not take into account flipping, which is crucial for single-shot localization. This is because, during single-shot training, some reverse sections remain unseen. We will elaborate on SSDA in the Settings section.

3.2 LiDAR Trajectories-coupled Interpolation

The primary motivations for LTI are twofold: (1) The training data for single-shot localization lacks trajectory diversity; (2) we can generate coupled trajectories by interpolating from a single trajectory. As analyzed in **Introduction**, question 2, and further validated in our experiments on **the Stability of Re-localization**. We summarize the process of LTI in the form of pseudocode, as shown below in Algorithm 1. LTI is a simple yet effective point cloud interpolation method. Its role is to simulate the movement of different trajectories. We explicitly generate two trajectories for left and right driving, rather than forward. We provide a comparative explanation in the experiment section titled **"Comparison with Other Single-shot Localization Solutions"**.

It can be broadly divided into three steps: (1) **Interpolate new poses from two consecutive frames**; In the interpolation of rotations, we employ quaternions to perform spherical linear interpolation (SLERP). First, we convert the rotation matrix to a quaternion:

$$q = a + b \cdot i + c \cdot j + d \cdot k \tag{1}$$

, a represents the angle of rotation, while (i,j,k) denote the three orthogonal axes in 3D space. Given the rotation angle, q can be represented as:

$$q = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})(b \cdot i + c \cdot j + d \cdot k) \tag{2}$$

After obtaining the quaternions corresponding to two rotation matrices, we can interpolate to get a new rotation pose:

$$q(\Delta t) = \frac{sin(1-\Delta t)\cdot\theta}{sin(\theta)} \cdot q_1 + \frac{sin(\Delta t \cdot \theta)}{sin(\theta)} \cdot q_2, \Delta t \in [0,1]. \tag{3}$$

$$\theta = arccos(q_1 \cdot q_2) \tag{4}$$

For the interpolation of translations, LTI utilizes effective linear interpolation (LERP). Therefore, we can obtain the interpolated rotation matrix $R_{\Delta t}$ and translation matrix $T_{\Delta t}$.

Algorithm 1: LTI Method Pseudocode

Input : Original point cloud trajectory $\mathcal{P} = \{P_1, ..., P_n\}$; rotation matrices $\mathcal{R} = \{R_1, ..., R_n\}$; translation vectors $\mathcal{T} = \{T_1, ..., T_n\}$

Output : New trajectory $\mathcal{P}_{\text{new}}$; interpolated rotations $R_{\Delta t}$; interpolated translations $T_{\Delta t}$

Data: R_p: rotation perturbation $\sim \mathcal{U}(-2°, 2°)$;
T_{py}: y-axis translation perturbation $\sim \mathcal{U}(-0.3, 0.3)$;
T_{px}: x-axis sine wave perturbation;
$\vec{V} = [0, \pm 2, 0]^T$: lateral shift vector;

for *each consecutive frame pair* $i \in \{1, ..., n-1\}$ **do**
  // Rotation interpolation via SLERP
  $R_{\Delta t} \leftarrow R_p \cdot \text{SLERP}(R_i, R_{i+1}, \Delta t)$;
  // Linear translation interpolation
  $T_{\Delta t} \leftarrow T_i + \Delta t \cdot (T_{i+1} - T_i)$;
  // Compute relative rotation and translation
  $R_r \leftarrow R_{\Delta t} \cdot R_i^\top$;
  $T_r \leftarrow R_r^\top \cdot (T_{\Delta t} - T_i)$;
  // Compute new point cloud
  $\mathcal{P}_{\text{new}} \leftarrow P_i \cdot R_r^\top - T_r$;
  // Shift y-coordinate of new points by ±2 and add perturbation for x and y coordinate
  for *each point* $p \in \mathcal{P}_{new}$ **do**
    $p.y \leftarrow p.y \pm 2$;
    $p.x \leftarrow p.x + T_{px}$;
    $p.y \leftarrow p.y + T_{py}$;
  end
  // compute new translation and add perturbation for new translation
  $T_{\Delta t} \leftarrow T_{\Delta t} \mp R_{\Delta t}\vec{V}$;
  $T_{\Delta t}.x \leftarrow T_{\Delta t}.x + T_{px}$;
  $T_{\Delta t}.y \leftarrow T_{\Delta t}.y + T_{py}$;
  return $(\mathcal{P}_{\text{new}}, R_{\Delta t}, T_{\Delta t})$;
end

(2) **Compute new point cloud.** In this step, we calculate the relative pose between the interpolated frame and the original point cloud, obtaining the relative rotation R_r and translation T_r. Then, we transform the point cloud to the position of the interpolated frame, resulting in a new point cloud $\mathcal{P}_{new}$.

(2) **Explicitly translate the point cloud and pose laterally**. We shift coordinates of y axis which is lateral direction for both points and poses. To simulate natural object motion or environmental noise as much as possible, We introduce random sine perturbations in the forward direction for both the point cloud and the pose, and apply uniform perturbations in the lateral directions. These perturbations help generate rich trajectory information.

3.3 Key Points Contrastive Learning

The motivation behind KP-CL is to enable the model to learn the correlation between different trajectories in the same scene. However, applying contrastive learning (CL) to large-scale point clouds

localization poses a challenge for GPU memory. The graph structure used in contrastive learning has far fewer nodes than point clouds. Unlike point cloud detection, where each frame contains many foreground objects of different categories that can easily form positive and negative sample pairs, point cloud localization does not have this advantage.

Therefore, we face two problems: (1) how to reduce the computational memory of CL, and (2) how to find positive and negative sample pairs. To address the first problem, we observed that the number of points is reduced to 1/8 of the original scene by the last layer of the encoder through convolution operations (see Figure.4). We refer to the points in the last layer as key points, whose deep features can effectively encode the scene and also serve as the input for KP-CL. Regarding the second problem, inspired by some related work[44] and considering that the features of the same scene on different trajectories are relatively similar, adding a small amount of noise for simulation is appropriate. Therefore, we construct positive and negative sample pairs by adding noise, without the need for multiple inferences through different data augmentations, which would cause an unbearable memory overhead.

Additionally, our KP-CL has a notable difference from the original CL. The original CL typically constructs a symmetric loss between augmented views. However, the original features are quite important in point cloud localization. Therefore, we compare the original features with the augmented views. Next, we present the process of KP-CL.

First, we augment the features x of the key points to obtain two views $\tilde{x^1}$, $\tilde{x^2}$, taking $\tilde{x^1}$ as an example:

$$\tilde{x^1} = x + sign(x) \cdot \frac{\varepsilon^1}{\left\|\varepsilon^1\right\|_2} \cdot 0.1 \tag{5}$$

ε is random noise which $\in$ [0,1]. Next, we project and normalize the features to obtain new features:$z, \tilde{z^1}, \tilde{z^2}$ (feature:128 dimensions). Finally, we define the contrastive learning loss. Through our experiments, we found that considering the diagonal elements of the similarity matrix between z and $\tilde{z^1}$ for positive samples, and considering both z with $\tilde{z^1}$ and $\tilde{z^2}$ for negative samples yields better results:

$$\mathcal{L}_{kp-cl} = -\frac{1}{N}\sum_{i=1}^{N}[\frac{z_i^T \cdot \tilde{z_i^1}}{\tau} - log(\sum_{\substack{j=1 \\ i\neq j}}^{N}(e^{z_i^T \cdot \tilde{z_j^1}/\tau} + e^{z_i^T \cdot \tilde{z_j^2}/\tau}))] \tag{6}$$

τ is the temperature parameter, and we set it as 0.15.

3.4 Ransac, Loss function and RFT

In the RANSAC section, we follow SGLoc[20] and utilize RANSAC to ultimately obtain the 6-DoF localization of the point cloud. The overall loss of PELoc is divided into two parts: localization loss and contrastive learning loss.

$$\mathcal{L}_{loc} = \frac{1}{P}\sum_{p_i \in P}\left\|p_i - p_i^*\right\|_1 \tag{7}$$

p_i denotes predicted position and p_i^* is the ground world coordinate position. Based on our experiments, we set the coefficient of the contrastive learning loss to 2. The total loss is:

$$\mathcal{L} = \mathcal{L}_{loc} + 2 \cdot \mathcal{L}_{kp-cl} \tag{8}$$

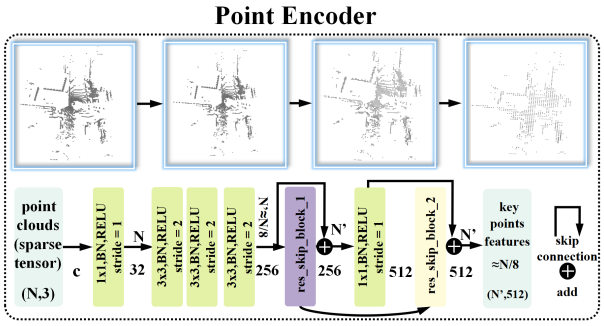


Figure 4: The structure of the PELoc encoder. We utilize the features of key points for KP-CL.

Additionally, we introduce a new training strategy called RFT (Remove 5% Random Consecutive Frames Training). RFT randomly discards 5% of consecutive frames in each training iteration. This approach aims to reduce the dependency on a single trajectory and enhance the diversity of the training trajectories.

4 Experiments

4.1 Settings

Benchmark datasets We conducted extensive experiments on three large-scale datasets: Oxford Radar RobotCar, Quality-enhanced Oxford and NCLT.

Oxford Radar RobotCar(Oxford)[2] is a large-scale urban autonomous driving dataset collected over a year using a specially equipped vehicle. It contains multi-sensor recordings from a 3D LiDAR, stereo cameras, Radar and GPS/INS system, capturing a 10km route through central Oxford under diverse conditions, including sunlight, rain, snow, and night-time illumination. The repeated traversals of the same nominal route exhibit natural trajectory variations characteristic of real-world driving scenarios, while maintaining consistent route topology. **Quality-enhanced Oxford** (QEOxford)[20], introduced in the SGLoc framework, addresses GPS/INS measurement errors in the original Oxford dataset through Iterative Closest Point (ICP)-based pose refinement. Specifically, it aligns raw data with high-quality reference submaps using ICP. We include QEOxford as an enhanced evaluation benchmark in our experiments. **NCLT[6]** was collected by the University of Michigan using a Segway robotic platform, spanning 27 sessions over 15 months across indoor/outdoor campus environments. It provides multimodal sensor data (omnidirectional vision, 3D LiDAR, IMU, RTK GPS) and captures significant environmental dynamics, including seasonal weather shifts, dense summer foliage, and it contains no single trajectory that covers all test paths. By fusing RTK GPS with LiDAR point cloud alignment, it delivers millimeter-level pose ground truth, making it ideal for evaluating localization robustness under real-world variability.

Setup and Training details We chose SGLoc as our baseline and improved the encoder, as shown in the Figure 4. The residual block structure can enhance the accuracy of the algorithm, which we call SGLoc-V2. Our GPU is an NVIDIA RTX 4090. Apart from the

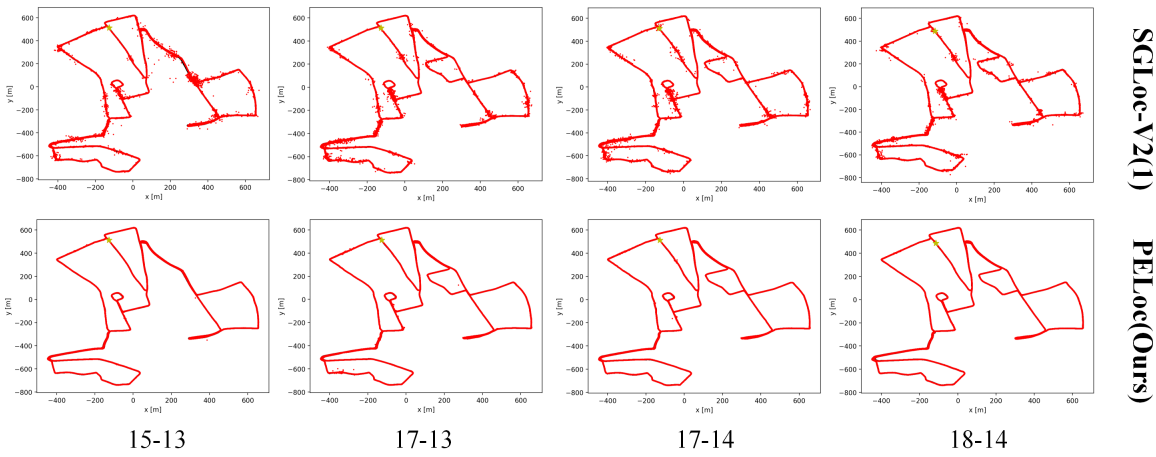


Figure 5: Visualization results of PELoc and SGLoc-V2 (single-shot, our baseline) on the four test trajectories of the QEOxford dataset. Black represents the ground truth, and red represents the predicted values.

Table 1: Comparison of average position error and orientation error between PELoc (single-shot) and SoTA, which uses four trajectories to train on the Oxford/QEOxford datasets. SGLoc-V2 (single-shot) serves as the baseline for PELoc. † denotes using four trajectories to train. (1) denotes using one original trajectory to train.

Oxford dataset(m,°) ↓								
Methods	PoseSOE†[47]	PosePN++†[48]	NIDALoc†[45]	DiffLoc†[19]	SGLoc-V2†[20]	LiSA†[40]	SGLoc-V2(1)	PELoc(ours (1)) SGLoc-V2-based
15-13-06-37	7.59/1.94	9.59/1.92	6.71/1.33	3.57/0.88	2.57/1.37	2.36/1.29	10.29/5.06	2.78/1.40
17-13-26-39	10.39/2.08	10.66/1.92	5.45/1.40	3.65/0.68	3.71/1.47	3.47/1.43	7.92/4.69	3.19/1.73
17-14-03-00	9.21/2.12	9.01/1.51	6.68/1.26	4.03/0.70	2.99/1.42	3.19/1.34	7.21/4.46	2.76/1.65
18-14-14-42	7.27/1.87	8.44/1.71	4.80/1.18	2.86/0.60	1.88/1.21	1.95/1.23	6.42/3.93	2.50/1.66
Average	8.62/2.00	9.43/1.77	5.91/1.29	3.53/0.72	2.78/1.36	2.74/1.32	7.96/4.54	2.81/1.61
Quality-enhanced(QE) Oxford dataset(m,°) ↓								
15-13-06-37	4.17/1.76	4.54/1.83	3.71/1.50	2.03/1.04	1.06/1.05	0.94/1.10	8.20/4.61	1.06/1.56
17-13-26-39	6.16/1.81	6.44/1.78	5.40/1.40	1.78/0.79	1.38/1.17	1.17/1.21	6.72/6.65	1.22/1.62
17-14-03-00	5.42/1.87	4.89/1.55	3.94/1.30	2.05/0.83	0.76/1.14	0.84/1.15	7.20/4.44	1.02/1.63
18-14-14-42	4.16/1.70	4.64/1.61	4.08/1.30	1.56/0.83	0.72/1.06	0.85/1.11	6.49/4.06	1.06/1.68
Average	4.98/1.79	5.13/1.69	4.28/1.38	1.86/0.87	0.98/1.11	0.95/1.14	7.15/4.94	1.09/1.62

Table 2: Comparison of average position error and orientation error between PELoc (single-shot) and SoTA, which use four trajectories to train on the NCLT datasets. SGLoc-V2 (single-shot) serves as the baseline for PELoc. † denotes using four trajectories to train. (1) denotes using one original trajectory to train.

NCLT dataset(m,°) ↓								
Methods	PoseSOE†[47]	PosePN++†[48]	NIDALoc†[45]	DiffLoc†[19]	SGLoc-V2†[20]	LiSA†[40]	SGLoc-V2(1)	PELoc(ours (1)) SGLoc-V2-based
2012-02-12	13.09/8.05	4.97/3.75	4.48/3.59	0.99/2.40	0.76/2.70	0.97/2.23	8.79/10.32	0.89/3.49
2012-02-19	6.16/4.51	3.68/2.65	3.14/2.52	0.92/2.14	0.71/2.49	0.91/2.09	37.12/17.48	1.22/3.73
2012-03-31	5.24/4.56	4.35/3.38	3.67/3.46	0.98/2.27	0.68/2.66	0.87/2.21	7.92/4.69	0.96/3.63
Average	8.16/5.71	4.33/3.26	3.76/3.19	0.96/2.27	0.71/2.61	0.92/2.18	17.94/10.83	1.02/3.61

network design, there are several notable differences between PELoc and SGLoc-V2. (1) SGLoc-V2 trained with four trajectories on both Oxford/QEOxford and NCLT datasets, whereas PELoc utilized only one. After careful consideration, we selected two relatively complete trajectories: 2019-01-11-14-02-26 for Oxford/QEOxford and 2012-02-18 for NCLT. (2) Given the abundance of samples four-trajectory training, data augmentation was not employed in SGLoc. However, for single-shot localization, data augmentation is crucial, prompting us to introduce SSDA.

Single-shot Data Augmentation: SSDA not only accounts for vehicle translation and angular offset but also considers the scenario where the vehicle travels in the opposite direction on a certain pass. The translation augmentation settings are as follows: the probability of translation along the xyz axes is 0.5 each, with translation ranges of [-3,3] m for the x and y axes and [-1,1] m for the z axis. The rotation augmentation settings are: the probability of rotation for roll, pitch, and yaw angles is 0.5 each, with rotation ranges of [$-\pi/4$,$+\pi/4$] for roll and pitch angles and [$-\pi/2$,$+\pi/2$] for the yaw angle. The reverse settings are: the probability of reversing is 0.5, with simultaneous flipping of the xy axes.

4.2 Comparison With State-of-the-Arts: Methods Using Four Trajectories

As far as we are aware, this is the first work on single-shot localization. In this section, we will primarily compare with the state-of-the-art methods that use four trajectories for localization. In subsequent sections, we will specifically compare with potential single-shot solutions. The algorithms we compare include the classics PoseSOE, PosePN++, PoseMinkLoc, and the latest state-of-the-art methods: FlashMix, SGLoc-V2, LiSA, and SGLoc-V2 (single-shot, our baseline). We will report the mean position and orientation errors.

Localization on the QEOxford/Oxford dataset: PELoc achieves an average position error of 2.81m and an orientation error of 1.61° on the four test trajectories of the Oxford dataset, reducing the errors of our single-shot baseline (7.96m, 4.54°) by 5.15m and 2.93° respectively. Due to the more inconsistencies among the trajectories in Oxford, our PELoc is able to almost match the performance of SGLoc-V2 (using four trajectories for training), with an average localization error lagging by only 3 centimeters. It is worth mentioning that PELoc achieved SoTA position error performance on the 17-13 and 17-14 trajectories. On the QEOxford dataset, thanks to the improved INS pose errors, PELoc attains better accuracy, with an average position error of 1.06m and an orientation error of 1.62°, a significant reduction from the single-shot baseline errors of 6.09m and 3.32°. The average position/orientation error lags behind that of SGLoc-V2 (using four trajectories for training) by 0.14m/0.51 degrees, which we consider to be an acceptable range. Additionally, we outperformed it on the 17-13 trajectory. The visualization Figure. 5 shows that PELoc exhibits minimal large jumps, indicating that it has learned robust environmental features from the limited one pass and can effectively handle challenges such as viewpoint changes and trajectory reversals.

Localization on the NCLT dataset: The NCLT dataset is more challenging for single-shot localization for longer temporal and spatial variations. The average error for single-shot localization (baseline) reaches 17.36 m and 12.17°. SGLoc-V2, which uses four trajectories for training, achieves state-of-the-art performance. PELoc also performs remarkably well, with an average error of 1.02 meters and 3.61 degrees. It achieves sub-meter position accuracy on the 02-12 and 03-31 trajectories. On the 02-19 trajectory, it reduces the baseline error by 33.16 m and 13.75°. However, PELoc fails to handle the 05-26 trajectory, we do not provide a comparison of accuracy. We will discuss this further in the discussion section. The visualization is presented in Figure.6.

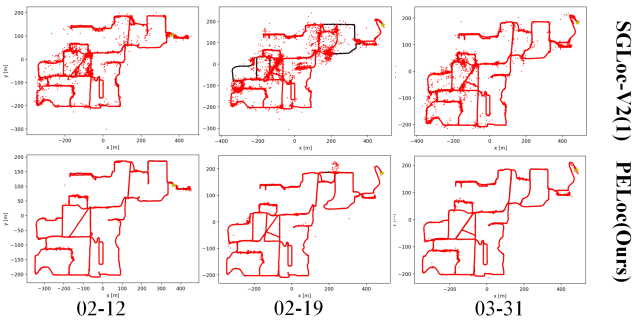


Figure 6: Visualization results of PELoc and SGLoc-V2 of the NCLT dataset.

4.3 Stability of Re-localization

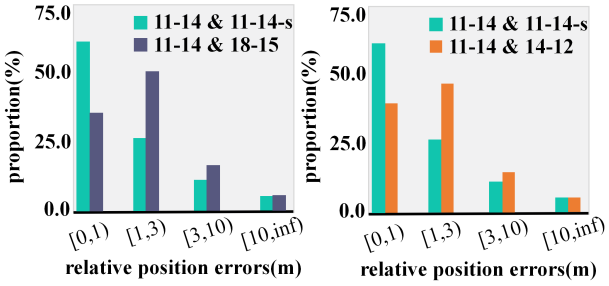


Figure 7: It shows the relative localization errors of the three trajectories in the same scene. The 11-14-s simulated point cloud corresponds to the 11-14 trajectory, while the 18-15,14-12 trajectory was collected on a different day.

PELoc's LTI exhibits a significant distinction from the original multi-trajectory training: the generated point cloud trajectories are coupled, whereas the other trajectories(e.g. 11-14 and 11-18) are affected by GPS/INS errors more or less. This coupling enhances the stability in re-localization. Taking the left picture as an example, we trained three models using the 11-14 trajectory from Oxford, and its simulated trajectory 11-14-s, as well as another trajectory, 18-15, for individual training. The three models were then tested on the 15-13 trajectory. By comparing the per-frame position errors of different models, we obtained the relative localization accuracy differences for the same scene. The Figure.7 illustrates that the relative position error between 11-14 and 11-14-s is primarily concentrated within 0-1 meters, whereas the relative position error between 11-14 and 18-15 is mainly distributed between 1-3 meters, indicating that the re-localization accuracy between the LTI trajectory and the original trajectory is higher. This approach eliminates additional GPS/INS errors between training trajectories. We believe this is beneficial for localization.

4.4 Comparison with Other Single-shot Localization Solutions

In the comparison with other single-shot localization solutions, we selected DTG and EF. The experimental results are shown in the

Table 3: Comparison with Other Single-shot Localization Method. DTG : Denser Trajectory Generation, EF: Equivariant Features. † means use the same SSDA.

single-shot Localization Solutions		
Methods	PE/OE (15-13) (m/°)	Training Epochs
baseline	8.20/4.61	50
DTG[25]†	2.89/2.57	50
EF[15]	2.69/1.94	100
Ours†	1.06/1.56	50

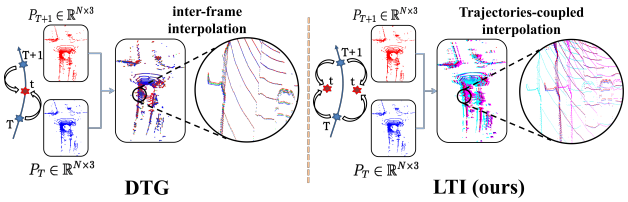


Figure 8: Comparison of point cloud generation between DTG (left) and LTI (right) methods. Both approaches utilize point clouds from frames T (blue) and $T+1$ (red) to synthesize the intermediate frame t (other colors).

Table3. DTG, inspired by point cloud interpolation works such as PointINet[25], aims to increase the density of trajectories (in the forward and backward directions, not lateral, see Figure.8). However, since PointINet and NeuralPCI[54] take nearly 1 minute to generate a single frame of point cloud, the time consumption is excessive. We used the method in LTI to simulate forward-moving point clouds, but the results were not as good as ours. The main reason is that the original point cloud trajectory is already dense, and the forward and backward interpolation methods cannot introduce a richer set of poses. The EF method, inspired by work[15], maps the point cloud to a canonical space and learns the equivariant rotation of the point cloud. Compared to the baseline, the effect is significant. However, it still falls short of our method. First, the LiDAR on Oxford dataset is mounted on the left side, which affects the learning of the reverse point cloud angles. Second, it requires more training epochs to fit both the original network and the equivariant network.

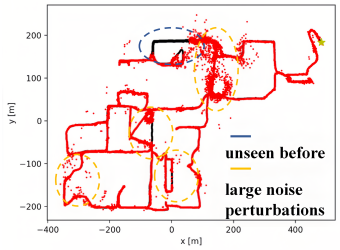
4.5 Ablation Study

The ablation is shown in Table 4. Ablation on SSDA: SSDA significantly boosts performance by effectively simulating trajectory rotations and translations, reducing the baseline error by 4.06m in position and 0.87 degrees in orientation. Ablation on LTI: The combination of LTI and SSDA serves as an antidote for single-shot localization. LTI's diversity in pose and point cloud further reduces errors by 2.49m in position and 1.85 degrees in orientation. Ablation on KP-CL: KP-CL enhances the model's ability to distinguish between similar scenes, lowering the baseline error by 2.51m in position and 1.65 degrees in orientation in standalone experiments. In the final combination, it reduces errors by an additional 0.45m in position and 0.33 degrees in orientation, effectively minimizing

Table 4: Performance Breakdown of PELoc on the QEOxford 15-13 Trajectory: KP-CL (Key Points Contrastive Learning), SSDA (Single-shot Data Augmentation), RFT (Remove of 5% random Consecutive Frames for Training), LTI (LiDAR Trajectories-coupled Interpolation).

KP-CL	SSDA	RFT	LTI	15-13 (m/°)
				8.20/4.61
✓				5.69/2.96
	✓			4.14/3.74
	✓	✓		4.00/3.77
	✓	✓	✓	1.51/1.89
✓	✓	✓	✓	1.06/1.56

localization noise. Ablation on RFT: RFT contributes to localization accuracy. As shown in Table 4, RFT reduces errors by 0.14m, this training strategy further enhances the diversity of trajectories.



Method	05-26
baseline	32.18m/22.53°
PELoc(Ours)	19.11m/11.32°

mean position/orientation error in NCLT 05-26 trajectory

Figure 9: Qualitative and quantitative analysis of the NCLT 05-26 trajectory

4.6 Discussion

Why LTI is effective: There are two main reasons for its effectiveness: (1) Though the trajectories generated by LTI aren't "real," the lateral interpolation operations in LTI lead the model to perceive that the car is on different trajectories. Thus, we posit that the coordinate offsets in the point cloud scenes can, to some extent, serve as a proxy for real point clouds. (2) The lateral interpolation and perturbation operations in LTI increase the diversity of poses and scenes.

Limitations of PELoc: PELoc is unable to handle the 05-26 trajectory in the NCLT dataset for two main reasons, as you can see in Figure .9: (1) The vegetation and trees are much denser on 05-26, resulting in significant differences in point cloud features compared to 02-18. (2) There are large sections of new trajectories on 05-26 that PELoc cannot process since it has not seen them before.

5 Conclusion

In this work, we present PELoc, a method for long-term LiDAR regression-based localization using a single trajectory, enabling near sub-meter localization accuracy on the QEOxford and NCLT datasets. Looking ahead, we commit to researching more long-term and robust one-pass localization algorithms.

Acknowledgments

We gratefully acknowledge the strong support of the ASC Laboratory at Xiamen University for this work. We also thank the reviewers and the Area Chair for their valuable suggestions. Additional experiments and discussions can be found on OpenReview.

References

[1] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. 2023. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2998–3007.

[2] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. 2020. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6433–6438.

[3] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. 2023. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5044–5053.

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. 2014. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 536–551.

[5] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. 2016. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3364–3372.

[6] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. 2016. University of Michigan North Campus long-term vision and lidar dataset. *The International Journal of Robotics Research* 35, 9 (2016), 1023–1035.

[7] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. 2024. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[8] Chao Fan, Zihan Li, Weike Ding, Huiming Zhou, and Kun Qian. 2024. Integrating artificial intelligence with SLAM technology for robotic navigation and localization in unknown environments. *International Journal of Robotics and Automation* 29, 4 (2024), 215–230.

[9] Raktim Gautam Goswami, Naman Patel, Prashanth Krishnamurthy, and Farshad Khorrami. 2024. FlashMix: Fast Map-Free LiDAR Localization via Feature Mixing and Contrastive-Constrained Accelerated Training. *arXiv preprint arXiv:2410.00702* (2024).

[10] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Hao Li, Ming Tang, and Jinqiao Wang. 2024. Filo: Zero-shot anomaly detection by fine-grained description and high-quality localization. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 2041–2049.

[11] Ting Han, Siyu Chen, Chuanmu Li, Zongyue Wang, Jinhe Su, Min Huang, and Guorong Cai. 2024. Epurate-net: Efficient progressive uncertainty refinement analysis for traffic environment urban road detection. *IEEE Transactions on Intelligent Transportation Systems* 25, 7 (2024), 6617–6632.

[12] Ting Han, Chenxi Du, Yijia Xie, Xinyan Xian, Xinchang Zhang, Bisheng Yang, and Yiping Chen. 2025. A 3D perspective for understanding the mechanisms of urban heat island and urban morphology using multi-modal geospatial data and interpretable machine learning. *Building and Environment* (2025), 113184.

[13] Xiangcheng Hu, Linwei Zheng, Jin Wu, Ruoyu Geng, Yang Yu, Hexiang Wei, Xiaoyu Tang, Lujia Wang, Jianhao Jiao, and Ming Liu. 2024. Paloc: Advancing slam benchmarking with prior-assisted 6-dof trajectory generation and uncertainty estimation. *IEEE/ASME Transactions on Mechatronics* (2024).

[14] Qingyuan Jia, Guiyang Luo, Quan Yuan, Jinglin Li, Congzhang Shao, and Ziyue Chen. 2023. MS-Transformer: Masked and Sparse Transformer for Point Cloud Registration. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 1375–1381.

[15] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. 2023. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*. PMLR, 15546–15566.

[16] Zaher M Kassas, Nadim Khairallah, Joe J Khalife, Chiawei Lee, Juan Jurado, Steven Wachtel, Jacob Duede, Zachary Hoeffner, Thomas Hulsey, Rachel Quirarte, et al. 2024. Aircraft navigation in GNSS-denied environments via radio SLAM with terrestrial signals of opportunity. *IEEE Transactions on Intelligent Transportation Systems* (2024).

[17] Bin Li, Mu Hu, Shuling Wang, Lianghao Wang, and Xiaojin Gong. 2021. Self-supervised visual-lidar odometry with flip consistency. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3844–3852.

[18] Wen Li, Chen Liu, Shangshu Yu, Dunqiang Liu, Yin Zhou, Siqi Shen, Chenglu Wen, and Cheng Wang. 2025. LightLoc: Learning Outdoor LiDAR Localization at Light Speed. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 6680–6689.

[19] Wen Li, Yuyang Yang, Shangshu Yu, Guosheng Hu, Chenglu Wen, Ming Cheng, and Cheng Wang. 2024. Diffloc: Diffusion model for outdoor lidar localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15045–15054.

[20] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. 2023. SGLoc: Scene geometry encoding for outdoor LiDAR localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9286–9295.

[21] Jing Ling, Zhengyang Wu, Changqing Li, Weisheng Li, Chao Zhang, and Yucheng Shu. 2025. Self-Geometry-Guided Direct Pose Regression Based on Dual Perspective Fusion for 2D-3D Cross Dimensional Spinal Surgery Navigation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[22] Haizhuang Liu, Junbao Zhuo, Chen Liang, Jiansheng Chen, and Huimin Ma. 2024. Affinity3D: Propagating Instance-Level Semantic Affinity for Zero-Shot Point Cloud Semantic Segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 9019–9028.

[23] Quan Liu, Hongzi Zhu, Zhenxi Wang, Yunsong Zhou, Shan Chang, and Minyi Guo. 2024. Extend your own correspondences: Unsupervised distant point cloud registration by progressive distance extension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20816–20826.

[24] Yujun Liu, Ruisheng Wang, Shangfeng Huang, and Guorong Cai. 2025. EdgeDiff: Edge-aware Diffusion Network for Building Reconstruction from Point Clouds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 17008–17018.

[25] Fan Lu, Guang Chen, Sanqing Qu, Zhijun Li, Yinlong Liu, and Alois Knoll. 2021. Pointinet: Point cloud frame interpolation network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 2251–2259.

[26] Pengcheng Shi, Shaocheng Yan, Yilin Xiao, Xinyi Liu, Yongjun Zhang, and Jiayuan Li. 2024. RANSAC back to SOTA: A two-stage consensus filtering for real-time 3D registration. *IEEE Robotics and Automation Letters* (2024).

[27] Weimin Shi, Changhao Chen, Kaige Li, Yuan Xiong, Xiaochun Cao, and Zhong Zhou. 2025. LangLoc: Language-Driven Localization via Formatted Spatial Description Generation. *IEEE Transactions on Image Processing* (2025).

[28] Marco Spanghero, Filip Geib, Ronny Panier, and Panos Papadimitratos. 2025. GNSS jammer localization and identification with airborne commercial GNSS receivers. *IEEE Transactions on Information Forensics and Security* (2025).

[29] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. 2020. Atloc: Attention guided camera localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10393–10401.

[30] Jingtao Wang and Zechao Li. 2024. 3DPCP-Net: A Lightweight Progressive 3D Correspondence Pruning Network for Accurate and Efficient Point Cloud Registration. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 1885–1894.

[31] Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. 2023. Hypliloc: Towards effective lidar pose regression with hyperbolic fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5176–5185.

[32] Sijie Wang, Rui She, Qiyu Kang, Xingchao Jian, Kai Zhao, Yang Song, and Wee Peng Tay. 2024. Distilvpr: Cross-modal knowledge distillation for visual place recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 10377–10385.

[33] Wei Wang, Bing Wang, Peijun Zhao, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, and Niki Trigoni. 2021. Pointloc: Deep pose regressor for lidar point cloud localization. *IEEE Sensors Journal* 22, 1 (2021), 959–968.

[34] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. 2024. DriveDreamer: Towards Real-World-Drive World Models for Autonomous Driving. In *European Conference on Computer Vision*. Springer, 55–72.

[35] Qiming Xia, Wenkai Lin, Haoen Xiang, Xun Huang, Siheng Chen, Zhen Dong, Cheng Wang, and Chenglu Wen. 2025. Learning to Detect Objects from Multi-Agent LiDAR Scans without Manual Labels. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 1418–1428.

[36] Qiming Xia, Wei Ye, Hai Wu, Shijia Zhao, Leyuan Xing, Xun Huang, Jinhao Deng, Xin Li, Chenglu Wen, and Cheng Wang. 2024. Hinted: Hard instance enhanced detector with mixed-density feature fusion for sparsely-supervised 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15321–15330.

[37] Yan Xia, Letian Shi, Zifeng Ding, Joao F Henriques, and Daniel Cremers. 2024. Text2loc: 3d point cloud localization from natural language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14958–14967.

[38] Rui Xu, Yong Luo, Han Hu, Bo Du, Jialie Shen, and Yonggang Wen. 2023. Rethinking the localization in weakly supervised object localization. In *Proceedings of the 31st ACM International Conference on Multimedia*. 5484–5494.

[39] Xuecheng Xu, Sha Lu, Jun Wu, Haojian Lu, Qiuguo Zhu, Yiyi Liao, Rong Xiong, and Yue Wang. 2023. Ring++: Roto-translation invariant gram for global localization on a sparse scan map. *IEEE Transactions on Robotics* 39, 6 (2023), 4616–4635.

[40] Bochun Yang, Zijun Li, Wen Li, Zhipeng Cai, Chenglu Wen, Yu Zang, Matthias Muller, and Cheng Wang. 2024. Lisa: Lidar localization with semantic awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15271–15280.

[41] Huan Yin, Xuecheng Xu, Sha Lu, Xieyuanli Chen, Rong Xiong, Shaojie Shen, Cyrill Stachniss, and Yue Wang. 2024. A survey on global lidar localization: Challenges, advances and open problems. *International Journal of Computer Vision* 132, 8 (2024), 3139–3171.

[42] Pengyu Yin, Haozhi Cao, Thien-Minh Nguyen, Shenghai Yuan, Shuyang Zhang, Kangcheng Liu, and Lihua Xie. 2024. Outram: One-shot global localization via triangulated scene graph and global outlier pruning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13717–13723.

[43] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. 2024. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20612–20622.

[44] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[45] Shangshu Yu, Xiaotian Sun, Wen Li, Chenglu Wen, Yunuo Yang, Bailu Si, Guosheng Hu, and Cheng Wang. 2023. Nidaloc: Neurobiologically inspired deep lidar localization. *IEEE Transactions on Intelligent Transportation Systems* 25, 5 (2023), 4278–4289.

[46] Shangshu Yu, Cheng Wang, Yitai Lin, Chenglu Wen, Ming Cheng, and Guosheng Hu. 2022. Stcloc: Deep lidar localization with spatio-temporal constraints. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2022), 489–500.

[47] Shangshu Yu, Cheng Wang, Chenglu Wen, Ming Cheng, Minghao Liu, Zhihong Zhang, and Xin Li. 2022. LiDAR-based localization using universal encoding and memory-aware regression. *Pattern Recognition* 128 (2022), 108685.

[48] Shangshu Yu, Cheng Wang, Zenglei Yu, Xin Li, Ming Cheng, and Yu Zang. 2021. Deep regression for LiDAR-based localization in dense urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 172 (2021), 240–252.

[49] Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. 2024. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9359–9367.

[50] Boming Zhao, Luwei Yang, Mao Mao, Hujun Bao, and Zhaopeng Cui. 2024. PNeRFLoc: Visual localization with point-based neural radiance fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 7450–7459.

[51] Jianwei Zhao, Yubo Wu, Wei Bao, Jingxiang Shi, and Xingguo Liu. 2025. Indoor Robot Mapping and Navigation System Based on Cyber-Physical Systems: Integration of SLAM Algorithm and Visual Information. *IEEE Transactions on Intelligent Transportation Systems* (2025).

[52] Shijia Zhao, Qiming Xia, Xusheng Guo, Pufan Zou, Maoji Zheng, Hai Wu, Chenglu Wen, and Cheng Wang. 2025. SP3D: Boosting Sparsely-Supervised 3D Object Detection via Accurate Cross-Modal Semantic Prompts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 29374–29384.

[53] Maoji Zheng, Ziyu Xu, Qiming Xia, Hai Wu, Chenglu Wen, and Cheng Wang. 2025. Seg2Box: 3D Object Detection by Point-Wise Semantics Supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 10591–10598.

[54] Zehan Zheng, Danni Wu, Ruisi Lu, Fan Lu, Guang Chen, and Changjun Jiang. 2023. Neuralpci: Spatio-temporal neural field for 3d point cloud multi-frame non-linear interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 909–918.

[55] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. 2023. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19370–19380.

[56] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4676–4687.

[57] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. 2021. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 16280–16290.