DiffLoc: Diffusion Model for Outdoor LiDAR Localization

Wen Li[1] Yuyang Yang[1] Shangshu Yu[2] Guosheng Hu[3] Chenglu Wen[1] Ming Cheng[1] Cheng Wang[1*]
[1] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University
[2] School of Computer Science and Engineering, Nanyang Technological University [3] Oosto

Abstract

Absolute pose regression (APR) estimates global pose in an end-to-end manner, achieving impressive results in learn-based LiDAR localization. However, compared to the top-performing methods reliant on 3D-3D correspondence matching, APR's accuracy still has room for improvement. We recognize APR's lack of robust features learning and iterative denoising process leads to suboptimal results. In this paper, we propose DiffLoc, a novel framework that formulates LiDAR localization as a conditional generation of poses. First, we propose to utilize the foundation model and static-object-aware pool to learn robust features. Second, we incorporate the iterative denoising process into APR via a diffusion model conditioned on the learned geometrically robust features. In addition, due to the unique nature of diffusion models, we propose to adapt our models to two additional applications: (1) using multiple inferences to evaluate pose uncertainty, and (2) seamlessly introducing geometric constraints on denoising steps to improve prediction accuracy. Extensive experiments conducted on the Oxford Radar RobotCar and NCLT datasets demonstrate that DiffLoc outperforms better than the state-of-the-art methods. Especially on the NCLT dataset, we achieve 35% and 34.7% improvement on position and orientation accuracy, respectively. Our code is released at https://github.com/liw95/DiffLoc.

1. Introduction

LiDAR localization is a crucial task for navigation planning, with a wide range of applications in computer vision, *e.g.*, autonomous driving [16], and augmented reality [6]. The goal of LiDAR localization is to take scanned point clouds as input and output its 6-DoF pose.

Contemporary state-of-the-art LiDAR localization methods are structure-based methods, which match points in the query point cloud to 3D world coordinates. Such 3D-3D correspondences are established either through scene coordinate regression [17] or feature matching [34, 41]. These correspondences are then used to estimate LiDAR pose by RANSAC [8]. However, these methods are usually costly in both time and memory [40]. Another approach is absolute pose regression (APR), which directly estimates global poses through a deep regression network without relying on preconstructed maps. Therefore, APR becomes favored due to its low computation and store cost [39].

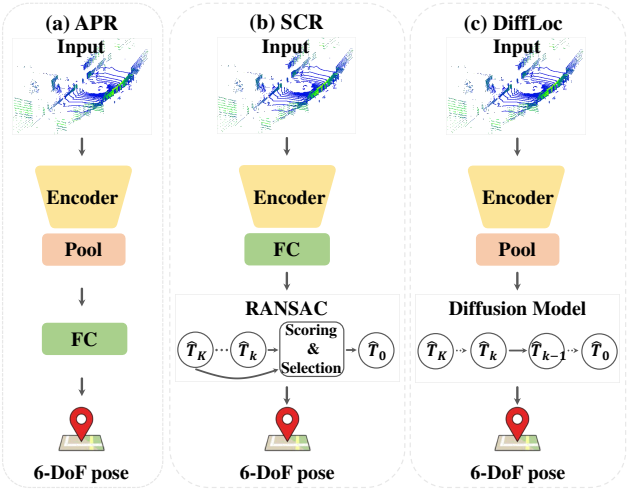


Figure 1. (a) Absolute pose regression (APR) directly estimates the poses in a forward pass. (b) Scene coordinate regression (SCR) predicts correspondences and applies iterative denoising to estimate poses. (c) DiffLoc introduces diffusion models to incorporate the same inherent spirit of iterative denoising into APR. Here, $\hat{T}_k$ represents the noisy pose of the k^{th} denoising step.

Despite the initial success, the localization accuracy of APR is still behind the 3D correspondence matching method. To illustrate this point, we analyze a recently proposed state-of-the-art method, SGLoc [17], to uncover the reasons. SGLoc is a LiDAR scene coordinate regression method, as shown in Fig. 1 (b). It decouples the localization process into two distinct stages: correspondence regression and pose estimation through RANSAC. (1) SGLoc emphasizes that APR lacks scene geometry features, which are robust and crucial for localization. (2) When comparing the fundamental differences between SGLoc and APR methods, we note that APR performs localization in

*Corresponding author.

Methods	Oxford Radar RobotCar [2]	NCLT [24]
SGLoc w/o RANSAC	22.04m/46.95°	24.58m/56.05°
SGLoc w/ RANSAC	3.14m/1.88°	1.83m/3.54°

Table 1. Ablation study of denoising process (RANSAC) in SGLoc on the Oxford Radar RobotCar [2] and NCLT [24] datasets. We report the mean error (m/°).

a single forward pass, as illustrated in Fig. 1 (a). It lacks the RANSAC-like iterative sampling component, which is commonly adopted for robust pose estimation.

For (1), to learn a discriminative feature, RangeViT [1] empirically demonstrates that employing foundation models in the RGB image domain can enhance LiDAR segmentation performance even with the large domain gap. Moreover, previous studies [11, 36] show that moving objects are harmful to learning geometrically robust features. This motivates us to learn foundation model derived robust features from static objects for LiDAR localization.

For (2), we conduct the ablation study of RANSAC in SGLoc, as shown in Tab. 1. The average accuracy improvement on position and orientation brought by RANSAC is 89.2% and 94.9%, respectively. These experiments prove the iterative sampling paradigm is critical for the structure-based method to achieve accurate results. Further, after analyzing the process of RANSAC, which iteratively removes outliers in noise data, it is actually a denoising process. Recently, denoising models, in particular, diffusion models [10], have demonstrated remarkable success in various tasks [4, 29, 49], *e.g.*, DiffusionDet [4] achieving object detection by taking it as a denoising process from noisy boxes to object boxes. It inspires us to investigate the diffusion model to formulate a denoising process for APR.

In this paper, we propose a novel framework, DiffLoc, which formulates LiDAR localization as a conditional generation of poses. We propose two novel designs that bridge the performance gap between APR and the structure-based methods. (1) We learn to encode point clouds to capture robust features. Derived from the foundation model (FM), we propose a static-object-aware pool (SOAP) to alleviate the impact from moving objects to learn robust features. (2) Motivated by RANSAC, we introduce iterative denoising with APR via a diffusion model conditioned on the features from (1). As shown in Fig. 1 (c), DiffLoc achieves localization by reversing the original poses from the noisy input progressively. In addition, due to the uniqueness of diffusion components, we drive two novel applications based on our modeling. (a) As DiffLoc can generate multiple plausible pose estimates by denoising different random noises, the variance of estimates is proposed to measure the pose uncertainty. (b) Due to the multi-step denoising nature of diffusion models, we introduce geometric constraints of relative poses on denoising steps to improve the accuracy of pose estimation.

Our contributions can be summarized as follows:

- Derived from the foundation model, DINO [25] in this work, we propose SOAP, which learns robust features and alleviates the negative impacts of moving objects.
- Motivated by RANSAC, we model LiDAR localization as a denoising process, from noisy to ground-truth poses, via a diffusion model, leading to significant performance improvements.
- Based on the uniqueness of diffusion models, including noise-driven inference and multi-step denoising, we propose to adapt our models to two additional applications: (a) evaluating pose uncertainty by aggregating multiple inferences and (b) enhancing the performance by geometric constraints to denoising steps.
- Extensive experiments on Oxford Radar RobotCar [2] and NCLT [24] datasets demonstrate the great effectiveness of our methods. In particular, we show that DiffLoc outperforms state-of-the-art methods by 35%/34.7% on the NCLT dataset.

2. Related work

2.1. Structure-based localization

Structure-based localization methods rely on 3D matching between the LiDAR point cloud and the 3D world coordinates to estimate the pose. These matches are established through feature descriptor matching [5, 15, 34, 41–43, 45] or regressing [17]. Descriptor-based methods can be further classified into retrieval-based [15, 34, 42, 43] and registration-based methods [5, 41, 45]. They need to store the point cloud descriptors for pose estimation. The recently proposed SGLoc [17] utilizes the network to regress the correspondences and uses RANSAC [8] for pose estimation, achieving state-of-the-art performance. However, since only the first stage is trainable, the method's accuracy is significantly affected by RANSAC.

2.2. Absolute pose regression

Absolute pose regression methods train CNNs to estimate the pose of input data, effectively encoding the scene through the network's parameters [3, 13, 14, 22, 31, 36, 38]. APR usually follows the same pipeline [30]. Specifically, they first use CNNs to learn the high-dimension feature description of scenes and then regress its poses.

Since LiDAR is robust to illumination change, LiDAR-based APR achieves impressive results on large-scale outdoor scenes. PointLoc [40] is the first LiDAR-based APR method, which encodes features by PointNet++ [27] followed by self-attention modules. The paper [47] proposes four methods with different feature learners. Some studies [46, 48] explore sequence constraints and achieve significant improvement. HypLiLoc [39] is the state-of-the-art APR method, which fuses multi-modal features in both

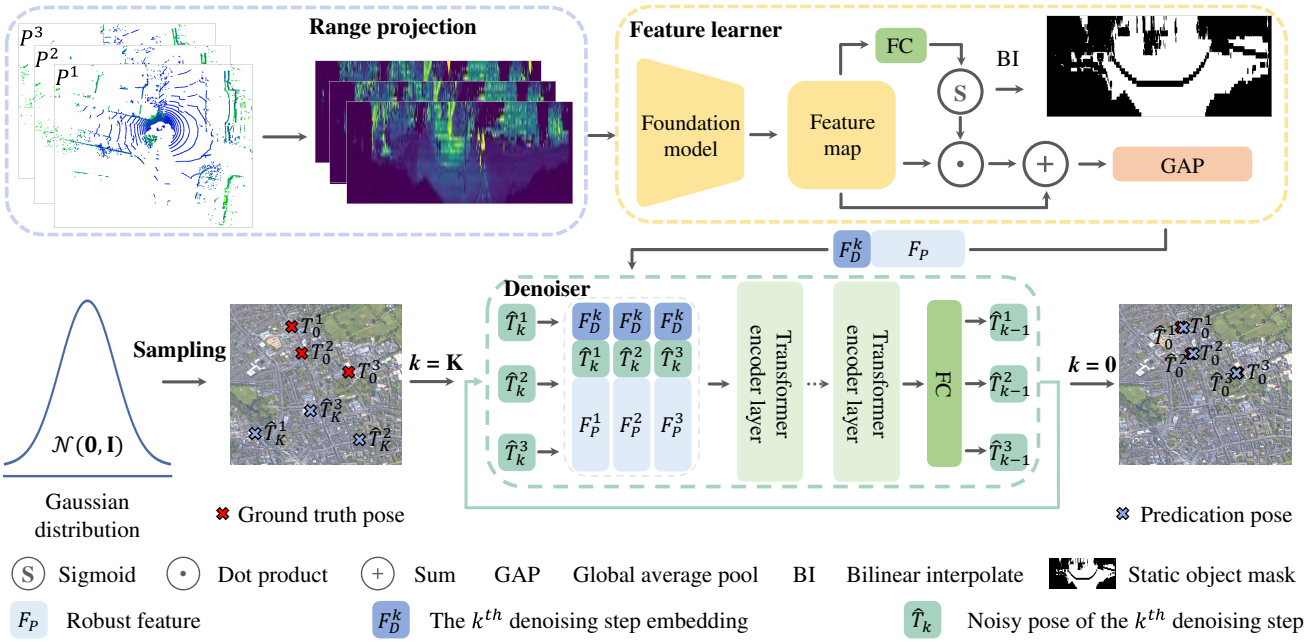


Figure 2. Illustration of DiffLoc framework during inference. First, N frame point clouds are projected into 2D images with range projection. Then, for the ith image, we use the foundation model and static-object-aware pool to learn the robust feature F_P^i. We also generate the denoising step embedding F_D^k for each k^{th} denoising step. Next, we sample N noisy poses $(\hat{T}_K^i)_{i=1}^N$ from Gaussian distribution and feed these to the denoiser K times to obtain the final poses, where the denoiser is also conditioned on F_P^i and F_D^k at each step, to obtain the prediction poses $(\hat{T}_0^i)_{i=1}^N$.

hyperbolic-Euclidean spaces, achieving promising results.

2.3. Diffusion models

Diffusion models are a type of deep generative model, which initiate from the random distribution and recover the data sample via a gradual denoising process. Specifically, in the training stage, Gaussian noise is gradually introduced to the ground truth sample. At the inference stage, diffusion models generate samples by reversing the original sample from the noisy input. They demonstrate remarkable results in many applications [4, 18, 20, 21, 23, 37, 49, 50]. In this paper, we recognize that diffusion models, given their iterative refinement properties, are especially suitable for modeling denoising processes for APR to bridge the performance gap between the structure-based methods.

3. Method

LiDAR-based APR shows impressive results in localization. However, its accuracy still lags behind structure-based methods due to its lack of robust feature learning and the iterative denoising process. In this paper, we propose DiffLoc, which utilizes the foundation model to learn robust features guided by static objects, and incorporate the iterative denoising process via a diffusion model, to bridge the performance gap (Sec. 3.1). Further, due to the uniqueness of diffusion models, we propose to adopt DiffLoc to two additional applications: (1) aggregating multiple inferences to evaluate pose uncertainty, and (2) seamlessly introducing geometric constraints on denoising steps to enhance the performance (Sec. 3.2).

3.1. DiffLoc

We now elaborate DiffLoc, as shown in Fig. 2, which can be divided into three key components. (1) A range projection preprocessor is employed to convert point clouds into images, preparing the input data for the foundation model. (2) A feature learner, derived from a foundation model DINO [25], learns to discriminatively encode the image. We propose a static-object-aware pool module, which can learn robust features. (3) A denoiser, actually a diffusion model conditioned on the input image encoded in Step (2), learns to recover the pose.

Range projection. To leverage the potent representation learning capabilities of the foundation model, we use the 2D range image representation for the input point clouds. Specifically, for the given tuple $\mathrm{P} = \left(P^i\right)_{i=1}^N$ of $N \in \mathbb{N}$ input point clouds with Cartesian coordinates (x, y, z), each point cloud is projected onto a range image as follows:

$$\begin{pmatrix} h \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\pi^{-1}\right] W \\ \left[1 - \left(\arcsin(z, r^{-1}) + |f_{down}|f_v^{-1}\right)\right] H \end{pmatrix}, \tag{1}$$

where (h, w) and (H, W) represent the pixel coordinates and size of the projected image, respectively. $f_v = f_{up} + f_{down}$ indicates the LiDAR sensor's vertical field-of-view. $r = \sqrt{x^2 + y^2 + z^2}$ signifies the range of each point. For generating the range image, we store (r, x, y, z, i) for each projected point. When multiple points are projected onto the same pixel, we retain only the features of the point with the smallest range. If a pixel has no points projected onto it, we fill it with zeros.

Feature learner. As mentioned earlier, APR methods cannot encode robust features effectively. This is an important reason for its suboptimal performance compared to structure-based methods [17]. Therefore, a robust feature learner is important for APR methods. Previous studies [1, 11, 28, 36] suggest that foundation models in images can improve the performance of point cloud tasks, inspiring us to investigate foundation models for LiDAR localization.

Firstly, to bridge the potential domain gap between range images and RGB images, we use the non-linear convolutional stem [44] to replace the embedding layer. Then, the output is fed to a ViT [7], which is used by DINO, to get the feature map $F \in \mathbb{R}^{M \times C}$, where M and C are the token number (without classification token) and feature dimension, respectively.

Subsequently, aiming to enhance the robustness over moving objects, we introduce a static-object-aware pool (SOAP) module. As shown in Fig. 2, the SOAP module takes the F as its input. We use the fully connected (FC) layer to squeeze the channel of F to 1 and employ the sigmoid operator to scale the feature to the range of 0 to 1. This process yields the static-object-aware attention mask, guiding the foundation model focus on static regions, *e.g.,* buildings. Then, we conduct the dot product and addition operation with F to achieve a robust feature. Finally, we use global average pooling (GAP) to obtain the global feature F_P. The output can be expressed as follows:

$$F_P = \mathrm{GAP}\left(F + \sigma\left(\mathrm{FC}\left(F\right)\right) \odot F\right), \tag{2}$$

where σ is the sigmoid function and $\odot$ denotes dot production. To ensure the attention mask generated appropriately emphasizes static objects, optimization during the training stage is conducted using pregenerated masks. Following SGLoc [17], we leverage the pretrained SPVCNN [33] to segment the moving objects.

Denoiser. As analyzed in the introduction section, another significant factor contributing to APR's suboptimal accuracy is its reliance on performing localization in a single forward pass, lacking the denoising process found in methodologies such as RANSAC. Therefore, drawing inspiration from diffusion models, we integrate denoising-like process in structure-based methods into APR.

DiffLoc models LiDAR localization as a denoising process from noisy poses to true poses via a diffusion model conditioned on encoded range images, which is achieved by our feature learner. During training, DiffLoc constructs a diffusion process represented by T_0, T_1, ..., T_K from the ground truth pose T_0 to a nearly pure noise T_K. Conversely, during inference, it constructs a denoising process represented by $\hat{T}_K$, $\hat{T}_{K-1}$, ..., $\hat{T}_0$ from a pure noise $\hat{T}_K$ to the pose prediction $\hat{T}_0$. The inference flowchart is shown in Fig. 2. We now elaborate on the training and inference process of DiffLoc.

During the training stage, the denoiser, *i.e.*, a diffusion model, is trained to learn the underlying distribution of LiDAR poses by recovering the ground truth pose from its corrupted version. Specifically, in each training iteration, a random diffusion step, denoted as $k \in \{1, 2, ..., K\}$, is chosen. With a predefined variance schedule $\beta_1, ..., \beta_k$, we introduce noises to the ground truth pose T_0 following the cumulative noise schedule, resulting in the noisy pose T_k.

$$\begin{aligned} q\left(T_k|T_0\right) &= \mathcal{N}\left(T_k; \sqrt{\overline{\alpha}_k}T_0, \left(1-\overline{\alpha}_k\right)\mathbf{I}\right), \\ T_k &= \sqrt{\overline{\alpha}_k}T_0 + \sqrt{1-\overline{\alpha}_k}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right), \end{aligned} \tag{3}$$

denoting $\alpha_k = 1 - \beta_k$ and $\overline{\alpha}_k = \prod_{i=1}^{k} \alpha_i$. Then, denoiser $\mathcal{D}_\theta$ predicts $\boldsymbol{\epsilon}$ to remove noise progressively. We implement the denoiser $\mathcal{D}_\theta$ by a transformer $\mathcal{T}$:

$$\mathcal{D}_\theta\left(T_k, k, \mathrm{P}\right) = \mathcal{T}\left(\left[T_k^i, F_D^k, F_P^i\right]_{i=1}^{N}\right), \tag{4}$$

where $[\,]$ denotes the concatenate operation, and the input of $\mathcal{T}$ is the sequence of noisy pose tuples T_k^i, unique step embedding F_D^k, and feature embedding F_P^i. F_D^k denotes the k^{th} denoising step generated via the sinusoidal function.

During the inference stage, the trained $\mathcal{D}_\theta$ can adapt to sequences with arbitrary noise levels. Therefore, we initiate the process with a pure noise sequence $\hat{T}_K$ and gradually reduce the noise. To accelerate the inference process, we utilize DDIM [32] to update the poses as:

$$\hat{T}_{k-1} = \sqrt{\overline{\alpha}_{k-1}}\left(\frac{\hat{T}_k - \sqrt{1-\overline{\alpha}_k}\mathcal{D}_\theta}{\sqrt{\overline{\alpha}_k}}\right) + \sqrt{1-\overline{\alpha}_{k-1}}\mathcal{D}_\theta, \tag{5}$$

where the $\hat{T}_{k-1}$ is sent into the denoiser $\mathcal{D}_\theta$ for the next step. This iterative denoising process will be repeated until reaching $\hat{T}_0$ at the end, which well approximates the underlying ground truth and is regarded as the final prediction.

Loss function. The output of DiffLoc consists of poses and static object masks. Therefore, the total loss function $\mathcal{L}$ should contain both denoising and segmentation components. Specifically, we employ L_1 loss to guide the model in predicting noise variable $\boldsymbol{\epsilon}$ as $\mathcal{L}_{\text{diff}} = \|\mathcal{D}_\theta\left(T_k, k, \mathrm{P}\right) - \boldsymbol{\epsilon}\|_1$, where the T_k and $\boldsymbol{\epsilon}$ are defined in Eq. (3). Binary cross-entropy loss is used to optimize the SOAP module $\mathcal{L}_{\text{seg}}$. The final loss is formulated as $\mathcal{L} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{seg}}$.

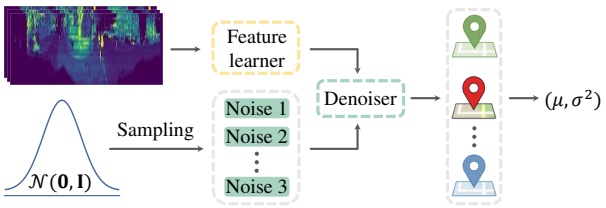


Figure 3. Workflow of the pose uncertainty evaluation.

3.2. Diffusion-derived Applications

The Diffusion model has unique modeling components, *e.g.*, random noise-driven inference, and multi-step denoising process. We delve deeper into investigating these unique components, based on which we propose to use this uniqueness for pose uncertainty evaluation and geometric constraint optimization.

Pose uncertainty evaluation. Previous efforts [11, 12] attempt to deactivate certain neurons during inference randomly, *i.e.*, Monte Carlo dropout [9], and then calculate the mean and variance through multiple inferences to estimate uncertainty. We show that DiffLoc can easily be implemented for pose uncertainty evaluation by aggregating multiple inferences without any other changes.

In the inference stage, the noisy pose is sampled from a standard distribution, leading to considerable variability between different runs of the inference method on the same inputs. Consequently, we can execute the inference multiple times and subsequently calculate the mean and variance for pose uncertainty evaluation, as illustrated in Fig. 3. Our experiments reveal a high correlation between the pose error and uncertainty (see Fig. 6).

Geometric constraint optimization. We show that DiffLoc can further increase the accuracy by leveraging geometric constraints between relative poses from the odometry, called geometric constraint-guided denoising (GCGD).

To this end, we use the relative poses from the odometry and guide denoising iterations so that the estimated poses satisfy the geometric constraints of relative poses. Specifically, suppose the $T^{i,j}$ denote the relative pose between two frames $\left(P^i, P^j\right)$ from the odometry, and denote $\left(\hat{T}^i, \hat{T}^j\right)$ the corresponding LiDAR poses. We can evaluate the geometric consistency by the relative pose $T^{i,j} \in \mathbb{R}$ error as:

$$e^{ij}\left(\hat{T}^i, \hat{T}^j, T^{i,j}\right) = \|T^{i,j} - \hat{T}^{i,j}\|_2, \tag{6}$$

where $\hat{T}^{i,j}$ is the relative pose between the LiDAR poses.

Next, we follow the guidance diffusion to guide the denoising to minimize relative pose error to satisfy the geometric constraints. In each denoising iteration, classifier guidance perturbs the predicted noise with a gradient of $\hat{T}_k$ conditioned guidance distribution $p\left(\mathrm{P}|\hat{T}_k\right)$:

$$\hat{\mathcal{D}}_\theta\left(\hat{T}_k, k, \mathrm{P}\right) = \mathcal{D}_\theta\left(\hat{T}_k, k, \mathrm{P}\right) - \sqrt{1-\overline{\alpha}_k}\nabla_{\hat{T}_k}\log p\left(\mathrm{P}|\hat{T}_k\right), \tag{7}$$

where $\hat{\mathcal{D}}_\theta$ replaces $\mathcal{D}_\theta$ for update the poses for the next step. Follow [37], $p\left(\mathrm{P}|\hat{T}_k\right)$ can be modeled as $p\left(\mathrm{P}|\hat{T}_k\right) \propto \prod_{i,j} \exp\left(-e^{i,j}\right)$.

4. Experiment

4.1. Experimental setup

Datesets and metrics. We evaluate DiffLoc for LiDAR localization on two large-scale outdoor benchmark datasets: Oxford Radar RobotCar [2] and NCLT [24]. As the evaluation metric, we use the mean position orientation error.

Oxford Radar RobotCar (Oxford) is an urban scene localization dataset [2], each trajectory spanning approximately 10km. It provides data from various sensors, including the LiDAR, camera, Radar, and GPS/INS. In our method, we use only LiDAR information. Oxford dataset encompasses diverse weather and traffic conditions, making it ideal for a comprehensive evaluation of the models.

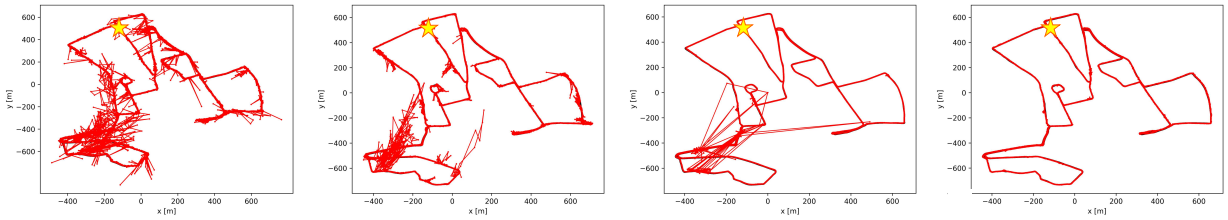
NCLT is a campus area localization dataset, each trajectory spanning about 5.5km. It consists of data from the LiDAR, omnidirectional camera, and GPS/INS. In our experiments, we use only LiDAR information. NCLT dataset covers a one-year data collection period, encompassing diverse seasonal environmental changes. It presents a range of scenarios, including both outdoor and indoor scenes with varying structural complexities, offering a robust evaluation for localization algorithms.

Implementation details. Our DiffLoc is implemented with Pytorch [26]. The foundation model DINOv2 [25] with ViT-S/16 backbone [7] is used for feature learning. For the denoiser, the Transformer [35] consists of 8 encoder layers with 4 attention heads for feature aggregation. The latent embedding dimension is 512. The range image size and patch size are set to $[32, 512]$ and $[4, 16]$, respectively. We define the total steps $K = 100$. We use a batch size of 28. The input point cloud sequence is a tuple of size 3 with a spacing of 2 frames. We train DiffLoc 150 epochs with the AdamW optimizer [7], which uses a single-cycle cosine annealing strategy [19] with a linear warm-up. The warm-up epoch and peak learning rate are set to 5 and $5e^{-4}$, respectively. We train the model using 4 RTX 3090 GPUs. On the Oxford dataset, we configure the iterative denoising steps to be 10. On the NCLT dataset, this value is set to 15.

Baselines and comparisons. To validate the performance of DiffLoc, we conduct a comparative analysis with several state-of-the-art learning-based LiDAR localization methods. For the structure-based localization, we chose PointNetVLAD (PNVLAD) [34] and DCP [41], which are

Methods	Mech.	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42	Average [m/°]	Ranks
PNVLAD [34]	S	18.14/3.28	24.57/3.08	19.93/3.13	15.59/2.63	19.56/3.03	12/11
DCP [41]	S	16.04/4.54	16.22/3.56	14.87/3.45	12.97/3.99	15.03/3.89	11/12
SGLoc [17]	S	**3.01**/1.91	**4.07**/2.07	**3.37**/1.89	**2.12**/1.66	**3.14**/1.88	**1**/6
PointLoc [40]	A	12.42/2.26	13.14/2.50	12.91/1.92	11.31/1.98	12.45/2.17	9/8
PosePN [47]	A	14.32/3.06	16.97/2.49	13.48/2.60	9.14/1.78	13.48/2.48	10/10
PosePN++ [47]	A	9.59/1.92	10.66/1.92	9.01/1.51	8.44/1.71	9.43/1.77	7/5
PoseMinkLoc [47]	A	11.20/2.62	14.24/2.42	12.35/2.46	10.06/2.15	11.96/2.41	8/9
PoseSOE [47]	A	7.59/1.94	10.39/2.08	9.21/2.12	7.27/1.87	8.62/2.00	6/7
STCLoc [46]	A	6.93/1.48	7.55/1.23	7.44/1.24	6.13/1.15	7.01/1.28	5/3
NIDALoc [48]	A	5.45/1.40	7.63/1.56	6.68/1.26	4.80/1.18	6.14/1.35	4/4
HypLiLoc [39]	A	6.88/**1.09**	6.79/**1.29**	5.82/**0.97**	3.45/**0.84**	5.74/**1.05**	3/**2**
DiffLoc (Ours)	A	**3.57**/**0.88**	**3.65**/**0.68**	**4.03**/**0.70**	**2.86**/**0.60**	**3.53**/**0.72**	**2**/**1**

Table 2. Oxford dataset comparison with the state-of-the-art methods. The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the **best** and **second-best** results.



(a) NIDALoc (7.63m, 1.56°) (b) HypLiLoc (6.79m, **1.29°**) (c) SGLoc (**4.07m**, 2.07°) (d) DiffLoc (**3.65m**, **0.68°**)

Figure 4. LiDAR localization results on the Oxford [2] dataset. The ground truth and prediction are black and red lines, respectively. The star denotes the first frame. The caption of each subfigure shows the mean position error (m) and orientation error (°).

the popular retrieval-based and registration-based methods. We also compare to SGLoc [17], which is a current state-of-the-art method in LiDAR localization. For the absolute pose regression, PointLoc [40], PosePN [47], PosePN++ [47], PoseMinkLoc [47], PoseSOE [47] and HypLiLoc [39] are chosen as comparison methods, which take a single-frame point cloud as input. Moreover, we also compare with multi-frame based APR, *e.g.*, STCLoc [46] and NIDALoc [48]. Note HypLiLoc is the current state-of-the-art LiDAR-based APR method.

4.2. Comparison with state-of-the-art methods

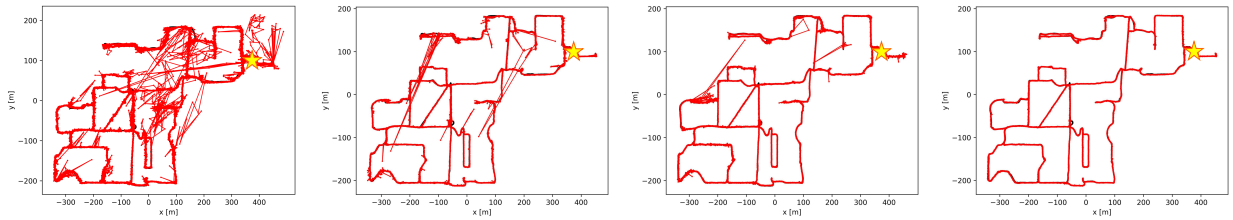
Results on the Oxford dataset. We first evaluate the proposed DiffLoc on the Oxford dataset, as shown in Tab. 1. We report the mean position and orientation error across all test trajectories and the respective ranking (where top-1 corresponds to the smallest error). Our method archives 3.53m/0.72° average error, which ranks second and first in position and orientation, respectively. Moreover, we get the best performance in 5 out of 8 metrics of four trajectories. Compared to the HypLiLoc, the LiDAR-based state-of-the-art APR method, DiffLoc improves by 38.5%/31.4%. Even when compared to a current state-of-the-art structure-based method, SGLoc, DiffLoc shows sufficient competitiveness. Specifically, although our positional accuracy is 12.4% behind, our orientation accuracy is significantly improved by 61.7%. Moreover, DiffLoc reduces the orientation error to within 1° on all trajectories.

Fig. 4 illustrates the trajectories predicted by the top 4 methods in Tab. 2 on 17-13-26-39 with mean position error (m) and orientation error (°). Compared to APR methods, *e.g.*, NIDALoc and HypLiLoc, DiffLoc gains significant improvements. The trajectories of SGLoc and DiffLoc closely align with ground truth. However, there are noticeable outliers in SGLoc, which renders its results unreliable in these regions. In contrast, DiffLoc offers a clean trajectory, indicating its capacity to produce more robust results.

Results on the NCLT dataset. We next test DiffLoc on the NCLT dataset. Tab 3 summarizes the results of all methods with mean position and orientation errors, and the ranking. Our method archives 1.19m/2.31° average error, which ranks the first compared to comparison methods, achieving the smallest position and orientation errors. Further, DiffLoc obtains the best performance in all metrics of four test trajectories. Compared to HypLiLoc, whose result is 1.95m/3.16°, DiffLoc gets a 39%/26.9% significant improvement. Even compared to SGLoc, our improvement is remarkable. Specifically, our results raise 35% and 34.7% on position and orientation, respectively. On the trajectory of 2012-05-26, DiffLoc outperforms SGLoc by a large margin, with a 50.7%/48.7% improvement. Moreover, DiffLoc reduces the error to the level of the sub-meter on all scenes

Methods	Mech.	2012-02-12	2012-02-19	2012-03-31	2012-05-26	Average [m/°]	Ranks
PNVLAD [34]	S	7.75/6.49	7.47/5.49	6.98/5.67	14.34/7.93	9.14/6.40	10/11
DCP [41]	S	9.84/6.84	8.27/5.16	8.94/5.96	15.62/7.99	10.67/6.49	12/12
SGLoc [17]	S	**1.20**/**3.08**	**1.20**/3.05	**1.12**/3.28	3.81/4.74	**1.83**/3.54	**2**/4
PointLoc [40]	A	7.23/4.88	6.31/3.89	6.71/4.32	10.02/5.32	7.57/4.60	8/7
PosePN [47]	A	9.45/7.47	6.15/5.05	5.79/5.28	13.47/7.77	8.72/6.39	9/10
PosePN++ [47]	A	4.97/3.75	3.68/2.65	4.35/3.38	9.59/4.49	5.65/3.57	6/5
PoseMinkLoc [47]	A	6.24/5.03	4.87/3.94	4.23/4.03	10.32/6.52	6.42/4.88	7/8
PoseSOE [47]	A	13.09/8.05	6.16/4.51	5.24/4.56	12.60/7.67	9.27/6.20	11/9
STCLoc [46]	A	4.91/4.34	3.25/3.10	3.75/4.04	8.67/5.23	5.15/4.18	5/6
NIDALoc [48]	A	4.48/3.59	3.14/2.52	3.67/3.46	6.60/4.56	4.47/3.53	4/3
HypLiLoc [39]	A	1.71/3.56	1.68/2.69	1.52/**2.90**	**2.90**/**3.47**	1.95/**3.16**	3/**2**
DiffLoc (Ours)	A	**0.99**/**2.40**	**0.92**/**2.14**	**0.98**/**2.27**	**1.88**/**2.43**	**1.19**/**2.31**	**1**/**1**

Table 3. NCLT dataset comparison with the state-of-the-art methods. The mechanism (Mech.) S/A denotes the structure-based localization/absolute pose regression. We highlight the **best** and **second-best** results.



(a) NIDALoc (4.48m, 3.59°) (b) HypLiLoc (1.71m, 3.56°) (c) SGLoc (**1.20m**, **3.08°**) (d) DiffLoc (**0.99m**, **2.40°**)

Figure 5. LiDAR localization results on the NCLT dataset [24]. The ground truth and prediction are black and red lines, respectively. The star denotes the first frame. The caption of each subfigure shows the mean position error (m) and orientation error (°).

(except 2012-05-26). Also, the orientation error is reduced to less than 3°. Results show DiffLoc can perform localization well on the NCLT dataset with mixed indoor and outdoor scenes and achieves state-of-the-art performance.

We visualize the trajectories predicted by the top 4 methods in Tab. 3 on 2012-02-12. As shown in this figure, the trajectory of DiffLoc is closer to the ground truth, and it has fewer wrong predictions (outliers). This further demonstrates the effectiveness of the proposed DiffLoc, which consistently produces robust localization results, even when dealing with the challenging NCLT dataset.

Results with pose uncertainty evaluation. We investigate the impact of DiffLoc on pose uncertainty evaluation on the 2012-05-26 trajectory of the NCLT dataset. Given the distinct differences between this trajectory and the training data, it contains more outliers, providing a suitable scene for exploring the relationship between estimated pose uncertainty and localization error. We initially generate final poses multiple times, each time with different initial noise poses. Subsequently, we aggregate these results, calculating both the mean and variance. The computed variance is utilized for pose uncertainty evaluation. Specifically, we summarize the inferences 10 times, illustrating the relationship between position error and variance, as shown in Fig. 6. It is evident that there is a high correlation between error and variance. It's important to note that

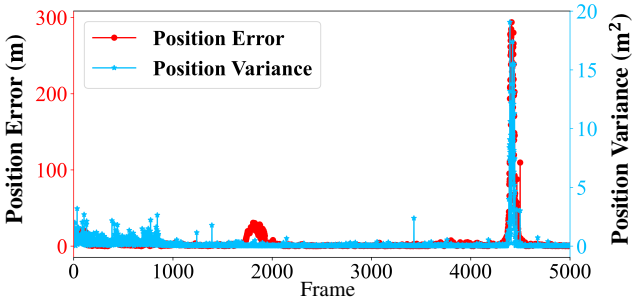


Figure 6. Pose uncertainty evaluation results. We show the relationship between position error and variance on the 2012-05-26 trajectory of the NCLT dataset.

only the denoiser requires multiple inferences, as the feature learner is deterministic. This experiment demonstrates that DiffLoc can effectively be employed for uncertainty evaluation, showcasing its extensibility.

Results with geometric constraint. We show that DiffLoc can further increase the accuracy via geometric constraint-guided denoising (GCGD). In this experiment, we use an SGD optimizer with the learning rate of $1e^{-3}$. To avoid spurious local minima, we apply this strategy to the last 5 denoising steps. During each step, we adjust the denoising by running 100 GCGD iterations. As illustrated in Tab. 4, DiffLoc with GCGD achieves an average error of 3.36m/0.62°, with an improvement of about 0.2m/0.1°.

Methods	15-13-06-37	17-13-26-39	17-14-03-00	18-14-14-42
DiffLoc	3.57m/0.88°	3.65m/0.68°	4.03m/0.70°	2.86m/0.60°
DiffLoc+GCGD	3.46m/0.79°	3.46m/0.59°	3.80m/0.58°	2.71m/0.53°

Table 4. Results with geometric constraint. **GCGD**: Geometric constraint-guided denoising.

	FM	SOAP	Denoising	Oxford (m/°)	NCLT (m/°)
1				6.64/1.55	7.11/8.06
2	✓			4.80/1.13	3.07/4.65
3	✓	✓		4.15/1.07	2.91/4.55
4	✓		✓	3.74/0.78	1.90/3.16
5	✓	✓	✓	3.53/0.72	1.19/2.31

Table 5. Ablation study on the Oxford and NCLT datasets. **FM**: Using the foundation model to learn features. **SOAP**: Using static-object-aware pool to guide feature reweight. **Denoising**: Using iterative denoising process to achieve localization.

This experiment demonstrates that DiffLoc can combine with odometry, leading to further improvements by incorporating relative pose constraints, showcasing its extensibility.

4.3. Ablation study

Study on FM. As shown in Tab. 5 between Row 1 and Row 2, using FM for feature learning brings a large improvement compared with the vanilla model without any proposed modules. On the Oxford dataset, it gains a 27.7%/27.1% increase on position and orientation accuracy, respectively. It even performs competitively with HypLiLoc, achieving a mean error of 4.80m/1.13° vs. 5.74/1.05°. On the NCLT dataset, FM also leads to a significant improvement. This study shows that harnessing FM pretrained on images can help improve LiDAR localization accuracy. This discovery is encouraging, as it suggests that we can accomplish model pretraining for LiDAR localization by leveraging a vast dataset of readily available, real crowdsourced maps.

Study on SOAP. We further conduct ablation experiments to demonstrate the importance of SOAP. On the Oxford and NCLT datasets, the comparison between Row 2 and Row 3 shows SOAP obtains an average improvement of 9.4% on position accuracy. Moreover, from Row 5 and Row 4, when our method with denoising module, SOAP still brings an average 0.46m/0.46° improvement. This shows that SOAP can further improve accuracy by guiding the learned feature reweighted by static objects.

Study on Denoising. The denoising results are reported in Row 4 and Row 5 in Tab. 5. Compared to Row 4 and Row 2, this module achieves a significant increase on the Oxford and NCLT datasets, with an average improvement of 30.1% on position and 31.5% on orientation accuracy. Moreover, compared to Row 5 and Row 3, it yields an average progress of 37%/41%. By using denoising, the average orientation error is reduced to within 1° and 3° on the Oxford and NCLT datasets, respectively. This significant improvement verifies the effectiveness of our modeling LiDAR localization as a denoising process.

Denosing Steps	Oxford (m/°)	NCLT (m/°)	Runtime (ms)
2	13.76/3.40	8.68/6.81	16
4	4.03/0.80	1.57/2.58	22
6	3.68/0.80	1.33/2.38	25
8	3.54/0.74	1.30/2.35	29
10	3.53/0.72	1.26/2.33	33
15	3.51/0.74	1.19/2.31	44
20	3.50/0.72	1.19/2.30	53

Table 6. Ablation study of the number of denoising steps.

Study on denoising steps. After training the model, the DiffLoc can adopt an arbitrary number of iterative denoising steps. To explore the impact of the number of iterative steps on the final performance, we experiment with different steps and report the results in Tab. 6. It is clear that more iteration steps result in better performance. We also report the time consumption of different steps. The optimal step is 10 as any more samples than it does not significantly improve performance. The running time at this step is 33ms, which is less than SGLoc with 38ms, achieving real-time performance. On the NCLT dataset, the optimal step is chosen as 15. The running time is about 44ms, much faster than SGLoc (75ms), also achieving real-time performance. Although compared to HypLiLoc (21ms), our DiffLoc is slower. However, in practice, the number of denoising steps can be adjusted flexibly without retraining the model. Specifically, when the step is 4, the running time of DiffLoc is only 22ms, outperforming HypLiLoc with 4.03m/0.80° vs. 5.74m/1.05° on the Oxford dataset, and 1.57m/2.58° vs. 1.95m/3.16° on the NCLT dataset.

5. Conclusion

In this paper, we explore and address the problem of lacking robust feature learning and iterative denoising process of APR. These are the most important factors for the performance gap between APR and structure-based methods. We propose a novel framework, DiffLoc, which treats LiDAR localization as a conditional generation of LiDAR poses. Specifically, for the first factor, we utilize the generalization capability of the foundation model for robust feature learning and propose SOAP to guide it to focus on the static objects in scenes. To solve the second challenge, we creatively incorporate an iterative denoising process to APR via a diffusion model. In addition, we show two attractive properties of DiffLoc, which can be used to evaluate pose uncertainty by aggregating multiple inferences and further improve the accuracy by utilizing geometric constraints. Extensive experiments demonstrate the effectiveness of our methods.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (No.62171393), the Fundamental Research Funds for the Central Universities (No.20720220064, No. 20720230033), and PDL (2022-PDL-12).

References

[1] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *CVPR*, pages 5240–5250, 2023. 2, 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. 2, 5, 6

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

[4] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *ICCV*, pages 19830–19843, 2023. 2, 3

[5] Zhi Chen, Kun Sun, Fan Yang, Lin Guo, and Wenbing Tao. Sc2-pcr++: Rethinking the generation and selection for efficient and robust point cloud registration. *IEEE TPAMI*, 2023. 2

[6] Yudi Dai, YiTai Lin, XiPing Lin, Chenglu Wen, Lan Xu, Hongwei Yi, Siqi Shen, Yuexin Ma, and Cheng Wang. Sloper4d: A scene-aware dataset for global 4d human pose estimation in urban environments. In *CVPR*, pages 682–692, 2023. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4, 5

[8] 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. 1, 2

[9] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *ICLR*, pages 1–12, 2016. 5

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2

[11] 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. 2, 4, 5

[12] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, pages 4762–4769. IEEE, 2016. 5

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

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

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

[16] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. Lo-net: Deep real-time lidar odometry. In *CVPR*, pages 8473–8482, 2019. 1

[17] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. Sgloc: Scene geometry encoding for outdoor lidar localization. In *CVPR*, pages 9286–9295, 2023. 1, 2, 4, 6, 7

[18] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *ICCV*, pages 10139–10149, 2023. 3

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 5

[20] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, pages 2837–2845, 2021. 3

[21] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *CVPR*, pages 12923–12932, 2023. 3

[22] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *CORL*, pages 1347–1356. PMLR, 2022. 2

[23] George Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas Guibas. Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *ICCV*, pages 14257–14267, 2023. 3

[24] Carlevaris-Bianco Nicholas, K. Ushani Arash, and M. Eustice Ryan. University of michigan north campus long-term vision and lidar dataset. *IJRR*, 35:545–565, 2015. 2, 5, 7

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 5

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 30, 2017. 2

[28] Chengyu Qiao, Zhiyu Xiang, Xinglu Wang, Shuya Chen, Yuangang Fan, and Xijun Zhao. Objects matter: Learning object relation graph for robust absolute pose regression. *Neurocomputing*, 521:11–26, 2023. 4

[29] Aimon Rahman, Jeya Maria Jose Valanarasu, Ilker Hacihaliloglu, and Vishal M Patel. Ambiguous medical image segmentation using diffusion models. In *CVPR*, pages 11536–11546, 2023. 2

[30] 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. 2

[31] Yoli Shavit, Ron Ferens, and Yosi Keller. Coarse-to-fine multi-scene pose regression with transformers. *IEEE TPAMI*, 2023. 2

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 4

[33] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, pages 685–702. Springer, 2020. 4

[34] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *CVPR*, pages 4470–4479, 2018. 1, 2, 5, 6, 7

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 5

[36] Bing Wang, Chaohao Chen, Chrisxiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. Atloc: Attention guided camera localization. In *AAAI*, pages 10393–10401, 2020. 2, 4

[37] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, pages 9773–9783, 2023. 3, 5

[38] Sijie Wang, Qiyu Kang, Rui She, Wee Peng Tay, Andreas Hartmannsgruber, and Diego Navarro Navarro. Robustloc: Robust camera pose regression in challenging driving environments. In *AAAI*, pages 6209–6216, 2023. 2

[39] Sijie Wang, Qiyu Kang, Rui She, Wei Wang, Kai Zhao, Yang Song, and Wee Peng Tay. Hypliloc: Towards effective lidar pose regression with hyperbolic fusion. In *CVPR*, pages 5176–5185, 2023. 1, 2, 6, 7

[40] 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. 1, 2, 6, 7

[41] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, pages 3523–3532, 2019. 1, 2, 5, 6, 7

[42] 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. 2

[43] Yan Xia, Mariia Gladkova, Rui Wang, Qianyun Li, Uwe Stilla, Joao F Henriques, and Daniel Cremers. Casspr: Cross attention single scan place recognition. In *ICCV*, pages 8461–8472, 2023. 2

[44] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *NeurIPS*, 34:30392–30400, 2021. 4

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

[46] Shangshu Yu, Cheng Wang, Yitai Lin, Chenglu Wen, Ming Cheng, and Guosheng Hu. Stcloc: Deep lidar localization with spatio-temporal constraints. *IEEE TITS*, 24(1):489–500, 2022. 2, 6, 7

[47] 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. 2, 6, 7

[48] Shangshu Yu, Xiaotian Sun, Wen Li, Chenglu Wen, Yunuo Yang, Bailu Si, Guosheng Hu, and Cheng Wang. Nidaloc: Neurobiologically inspired deep lidar localization. *IEEE TITS*, 2023. 2, 6, 7

[49] Lukas Zbinden, Lars Doorenbos, Theodoros Pissas, Adrian Thomas Huber, Raphael Sznitman, and Pablo Márquez-Neila. Stochastic segmentation with conditional categorical diffusion models. In *ICCV*, pages 1119–1129, 2023. 2, 3

[50] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, pages 5826–5835, 2021. 3