RCP-LO: A Relative Coordinate Prediction Framework for Generalizable Deep LiDAR Odometry

Chen Liu[1,2*], Wen Li[1,2*], Yongshu Huang[1,2], Minghang Zhu[1,2], Yuyang Yang[1,2], Dunqiang Liu[1,2], Sheng Ao[1,2], Cheng Wang[1,2†]

[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China
[2]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, School of Informatics, Xiamen University, China
{chenliu, liwen777}@stu.xmu.edu.cn, cwang@xmu.edu.cn

Abstract

LiDAR odometry is a critical component of SLAM in autonomous driving and robotics. Learning-based methods have shown remarkable performance by regressing relative poses in an end-to-end manner. However, when applying these trained models, originally developed on the widely used KITTI dataset, to other scenes, performance often drops significantly. In other words, existing methods struggle to generalize well to new environments. To address this challenge, we propose RCP-LO, a simple yet effective LiDAR odometry framework. We introduce a novel representation for relative poses, reformulating them as relative coordinates, which can then be solved using geometrical verification. This approach avoids overly simplified pose representations and makes better use of scene geometry, thereby improving generalization. Moreover, to capture the inherent uncertainties in relative pose estimation from occluded LiDAR point clouds from dynamic environments, we adapt our framework to learn a denoising diffusion model, allowing for sampling plausible relative coordinates while improving robustness. We also introduce a differentiable geometric weighted singular value decomposition module, enabling efficient pose estimation through a single forward pass. Extensive experiments demonstrate that RCP-LO, trained exclusively on the KITTI dataset, achieves competitive performance compared to SOTA learning-based methods and generalizes effectively to the KITTI-360, Ford, and Oxford datasets.

Code — https://github.com/lcxmu/RCP-LO

Introduction

LiDAR odometry aims to estimate the relative pose transformation between two frames of a moving platform by analyzing LiDAR point cloud data. This task is a fundamental component of many applications, *e.g.*, autonomous driving (Liu et al. 2023a; Wang et al. 2021b) and robotics (Segal, Haehnel, and Thrun 2009).

LiDAR odometry methods are broadly categorized into two categories: traditional geometry-based methods (Segal, Haehnel, and Thrun 2009; Zhang, Singh et al. 2014; Dellenbach et al. 2022; Vizzo et al. 2023) and learning-based methods (Li et al. 2019; Zheng et al. 2020; Xu et al. 2022a; Ali et al. 2023). The former rely on hand-crafted feature extraction from raw point clouds, whereas the latter leverage deep neural networks to extract robust features and directly regress 6-DoF relative poses. Recent studies (Liu et al. 2019; Hu et al. 2020) find that learning-based methods are effective at handling sparse point clouds and dynamic environments, which are usually difficult for traditional methods. Consequently, learning-based methods have garnered substantial attention in recent research.

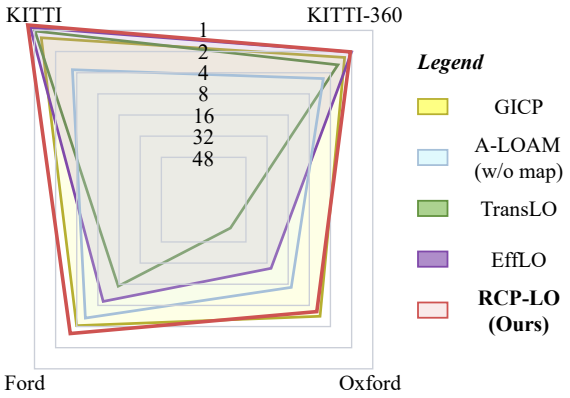

Figure 1: **Generalization Results.** We report the average translational RMSE (%) on the KITTI, KITTI-360, Ford, and Oxford datasets. The deep learning-based methods are trained only on KITTI and directly tested on the other datasets without any further training or fine-tuning.

Despite the initial success, we observe a significant drop in performance when applying a trained learning-based method to other scenes. As shown in Fig. 1, we evaluate two representative learning-based methods, TransLo (Liu et al. 2023a) and EfficientLO (EffLO) (Wang et al. 2022), trained on KITTI dataset (Geiger et al. 2013), testing on KITTI-360 (Liao, Xie, and Geiger 2022), Ford (Pandey, McBride, and Eustice 2011) and Oxford (Barnes et al. 2020) datasets. For comparison, we also test classical methods GICP (Segal,

*These authors contributed equally.
†Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Haehnel, and Thrun 2009) and A-LOAM (w/o map) (Zhang, Singh et al. 2014). On the KITTI-360 dataset, which is collected using the same LiDAR sensor and platform in the same city, EffLO outperforms GICP. However, in the cases of Ford (same LiDAR, different platform and city) and Oxford (different LiDAR, platform and city), the performance of EffLO decreases significantly. These results show that the generalization of existing learning-based methods have room to improve.

Existing learning-based methods usually follow a similar pipeline. Specifically, they first use a CNN to learn global feature descriptors between two frames of point clouds, and then regress their relative poses directly. However, a recent study (Zhang et al. 2024b) suggests that directly regressing overly simplified pose representations $T = [R, t]$, where $t \in \mathbb{R}^3$ is a translation vector and $R \in \mathbb{R}^r$ represents the orientation (*e.g.,* a 4D unit quaternion, a 3D Euler angle or a matrix), using CNNs results in suboptimal pose estimation accuracy and limited generalization. This motivates us to explore an alternative pose representation for LiDAR odometry to solve this problem.

Inspired by recent works (Li et al. 2023; Liu et al. 2023b), in this paper, we propose to reformulate the task of relative pose regression as relative coordinate prediction. Given two consecutive point clouds, P_1 and P_2, our method predicts the coordinates of each point in P_2 under the local system of P_1, instead of directly regressing the relative pose. This approach helps preserve the spatial geometric structure of the point cloud, which is crucial for achieving accurate pose estimation (Sattler et al. 2019; Li and Wang 2020). Moreover, the coordinates of the same point cloud under different reference frames implicitly represent the relative transformation between the two local coordinate systems. Therefore, the relative pose can then be solved by a robust estimator such as RANSAC (Fischler and Bolles 1981).

Following previous works (Wang et al. 2021b; Liu et al. 2023a), we employ a hierarchical network to progressively predict the relative coordinates of P_2 from coarse to fine. Considering the ambiguities due to partial observations in dynamic environments, we use a probabilistic diffusion model (Ho, Jain, and Abbeel 2020) to achieve relative coordinates prediction, and find that this further improves the performance. In addition, we realize RANSAC introduces challenges in efficient end-to-end training and inference due to its iterative processing on CPUs (Wu et al. 2022). Therefore, we introduce a differentiable Geometric Weighted Singular Value Decomposition (GW-SVD) module, which learns the confidence weights and is further refined by integrating geometric consistency-based measures, achieving pose solved by once forward.

We conduct extensive experiments on the KITTI, KITTI-360, Ford, and Oxford datasets. The results demonstrate that our method outperforms existing learning-based methods in accuracy and generalization, as shown in Fig. 1.

Our contributions can be summarized as follows:

- We introduce a new paradigm by reformulating relative pose regression in LiDAR odometry as relative coordinate prediction, preserving the point cloud's geometric structure for superior accuracy and generalization.
- Leveraging a probabilistic diffusion model for robustness against partial observations and a differentiable GW-SVD module, we achieve efficient, single-pass pose estimation with geometric consistency.
- Trained solely on KITTI, our method achieves performance comparable to SOTA learning-based odometry methods on the KITTI benchmark, while demonstrating remarkable cross-domain generalization capabilities on KITTI-360, Ford, and Oxford.

Related Works

LiDAR Odometry

Based on the implementation approach, existing methods can be categorized into traditional geometry-based methods (Zhang, Singh et al. 2014) and learning-based methods (Wang et al. 2022).

Traditional geometry-based methods rely on handcrafted operators for feature extraction and matching in point clouds. ICP (Chetverikov et al. 2002; Arun, Huang, and Blostein 1987) minimizes alignment error via iterative matching and transformation estimation. GICP (Segal, Haehnel, and Thrun 2009) adds probabilistic modeling for better geometric uncertainty handling. VGICP (Koide et al. 2021) further accelerates inference by Voxelized representation. LOAM (Zhang, Singh et al. 2014) enables precise localization and mapping by extracting edge and planar features, while LeGO-LOAM (Shan and Englot 2018) incorporates multi-sensor fusion, and F-LOAM (Wang et al. 2021c) refines the pipeline into a non-iterative, two-stage process for improved efficiency. However, its accuracy and robustness still face significant challenges in scenarios with sparse point clouds and dynamic object interference.

Leveraging the robust feature extraction capability of CNN, learning-based methods have achieved impressive performance. LO-Net (Li et al. 2019) employs 2D CNN and projected point cloud images, achieving end-to-end odometry for the first time. EffLO (Wang et al. 2021b, 2022) introduces a hierarchical embedding mask to optimize pose estimation, significantly improving localization accuracy. TransLO (Liu et al. 2023a) designs a window-based transformer module to strengthen global features and regress relative pose estimation. DELO (Ali et al. 2023) is a notable advancement, incorporating partial optimal transport of LiDAR descriptors and predicting uncertainty for robust pose estimation. On the other hand, DSLO (Zhang et al. 2024a) enhances accuracy and computational efficiency by introducing an inconsistent spatio-temporal propagation method. DVLO (Liu et al. 2025) improves upon this by proposing a deep visual-LiDAR odometry approach that leverages local-to-global feature fusion and bi-directional structure alignment. However, we find that existing deep learning-based methods perform well within the domain of the training data, but struggle to generalize to other datasets with different scenes or LiDAR sensor types, as shown in Fig. 1. This motivates us to explore a more generalized method.

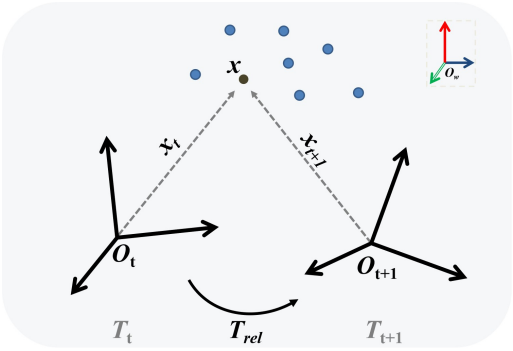

Figure 2: **Relative coordinates representation.** T_t and T_{t+1} denote the global pose at times t and $t+1$. T_{rel} is the relative transformation from t to $t+1$. x is the co-visible point at both times, which is represented as x_t and x_{t+1} in the local coordinate systems at time t and $t+1$, respectively.

Point Cloud Coordinates Prediction

Similar to image-based methods (Park, Patten, and Vincze 2019; Li et al. 2020; Wang et al. 2024), recent works (Li et al. 2023; Lu et al. 2021) have explored coordinate prediction for LiDAR pose estimation. Existing methods can be categorized into scene coordinate prediction and relative coordinate prediction, addressing global and relative pose estimation, respectively. SCR (Li et al. 2023; Yang et al. 2024) build coordinates using the input point cloud and global ground-truth pose. They use a network to predict the corresponding scene coordinate. However, these methods are scene-dependent and struggle to generalize to new scenes. HRegNet (Lu et al. 2021) enhances similarity feature descriptors to accurately identify candidate keypoints and employs a weighted aggregation strategy to generate highly reliable sparse virtual relative points. However, this approach heavily depends on the weighted averaging of neighboring points in the point cloud. In contrast, our method directly predicts the full set of relative coordinates by leveraging learned local features and association embeddings from two-frame point clouds.

Method

Relative Coordinates Representation

As shown in Fig. 2, two point clouds are acquired at times t and $t+1$, along with their corresponding global sensor poses, T_t and T_{t+1}, respectively. The relative transformation from time t to $t+1$ is denoted as $T_{rel} = [R_{rel}, t_{rel}]$, this transformation satisfying:

$$T_{t+1} = T_t \cdot T_{rel}. \tag{1}$$

To define the relative coordinate, we assume a point x in space that can be observed in both point clouds at times t and $t+1$. Specifically, this point is represented as x_t in the local coordinate system at time t (with the LiDAR sensor as the origin), and as x_{t+1} at time $t+1$, satisfying:

$$T_t \cdot x_t = T_{t+1} \cdot x_{t+1}. \tag{2}$$

Therefore, theoretically there exists point pairs satisfying $x_t = T_{rel} \cdot x_{t+1}$ between two timestamps. However, the inherent sparsity, disorder, and occlusion issues of LiDAR point clouds make it difficult to find correct corresponding point pairs. Rather than seeking direct point-to-point correspondences between point clouds at different timestamps, we estimate the transformed coordinates of x_{t+1} in the local coordinate system O_t of the previous frame's point cloud:

$$x'_{t+1} = T_{rel} \cdot x_{t+1}. \tag{3}$$

Given the same set of points observed in different local coordinate systems, the ego-motion transformation T_{rel} can be estimated between them. Note that x'_{t+1} and x_t have no direct correspondence due to point cloud disorder. The relative coordinate (RC) can then be expressed as:

$$RC = x'_{t+1} - x_{t+1} = T_{rel} \cdot x_{t+1} - x_{t+1}. \tag{4}$$

In this paper, given two consecutive point clouds, P_1 and P_2, we aim to predict the RC using a neural network. By establishing point-wise correspondences between P_2 under different local frames, we obtain an implicit representation of the relative pose, which can then be used to robustly estimate the relative transformation using RANSAC (Fischler and Bolles 1981).

Framework

We now introduce the proposed method, RCP-LO, as shown in Fig. 3, which can be divided into three components. (1) Feature extraction: This step generates deep descriptors for consecutive point clouds. (2) Diffusion-based relative coordinates prediction: Using the input point clouds encoded in Step (1), this step learns to predict the relative coordinates. (3) Geometric Weighted Singular Value Decomposition (GW-SVD): This component solves for the pose based on the input points and predicted relative coordinates. (4) Pose refinement and relative coordinate adjustment: A hierarchical update process for refining both the predicted pose and relative coordinate.

Feature Extraction. In this stage, we use PointConv (Wu, Qi, and Fuxin 2019) to encode the local coordinates of neighboring points and aggregate their local features, resulting in point-level features. Finally, point coordinates and features are output at $L = 4$ different scales.

After obtaining the multi-scale point coordinates, denoted as P^l, we define the target point cloud as $P^l_{gt} = T_{gt} \cdot P^l_2$. This target point cloud will later be used to calculate the ground-truth relative coordinates at each level RC^l_{gt}, as described in Eq. 10, which will be explained in the following sections.

Diffusion-based Relative Coordinates Prediction. Relative coordinates provide an effective representation for pose estimation. However, in a dynamic environment, *e.g.,* an autonomous driving environment, the presence of dynamic objects can cause occlusions in the captured point clouds, introducing inherent uncertainty into the prediction task. To explicitly model such uncertainty, we we employ a Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020).

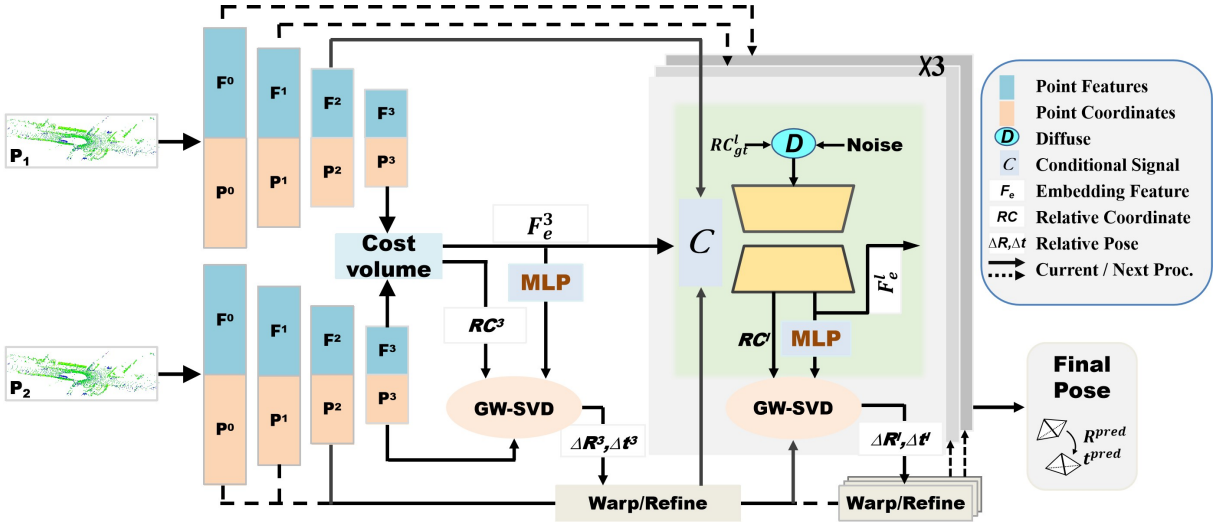

Figure 3: **The details of the proposed architecture.** Given the input point clouds $P_1, P_2 \in \mathbb{R}^{N\times 3}$, we first extract multi-scale features F^l using PointConv. Initial relative coordinates RC^3 are predicted via a cost-volume module and further refined by the diffusion module. Both initial and refined coordinates are decomposed with GW-SVD to progressively estimate the final pose.

DDPM is a denoising-based generative framework that perturbs data by adding Gaussian noise in a forward process and learns to remove this noise step-by-step in a reverse process. This enables the model to recover clean predictions even when observations are incomplete or noisy.

In the forward process, given the ground-truth relative coordinates RC^l_{gt}, the noisy version at diffusion k is:

$$RC^l_{noisy}(k) = \sqrt{\overline{\alpha}(k)}RC^l_{gt} + \sqrt{1-\overline{\alpha}(k)}\,\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{5}$$

where $\overline{\alpha}(k) = \prod_{i=1}^{k}(1-\beta(i))$, and $\beta(1), ..., \beta(k)$ is the predefined variance schedule.

The diffusion model predicts clean coordinates from noisy inputs and is trained with an L_2 regression loss:

$$\ell^l_{res} = \left\| RC^l_{gt} - RC^l_{pred} \right\|_2^2, \tag{6}$$

here RC^l_{pred} is the prediction of the diffusion model.

During inference, following previous works (Tevet et al. 2022; Liu et al. 2024), we generate a prediction from the condition and randomly sampled noise. The condition, denoted as C^l, consists of the concatenation of three features: geometric features extracted from P^l_2, the cross-frame cost volume (Wang et al. 2021a), and the local coordinate embedding generated by RC^l_{pred} through MLPs.

Geometric Weighted SVD. To achieve end-to-end optimization and single-pass pose estimation, we propose the GW-SVD module.

We now introduce the input of GW-SVD, as shown in Fig. 3. At the coarse level ($l = 3$), GW-SVD takes the point cloud P^3_2, the predicted coarse relative coordinates RC^3_{pred} obtained by cost-volume, and the feature-based weights W^3_f generated by feature embedding F^3_e through MLPs. Here, F^3_e denotes correlation embedding features between the two frames, generated by cost-volume, and W^3_f represents the accuracy of each predicted point. Then, F^3_e is propagated to the next layer ($l = 2$) through the set-upconv layer (Liu, Qi, and Guibas 2019) to upsampling, making the point number equal to P^2_2. At the fine level ($l \leq 2$), instead of cost-volume, we use a diffusion module to generate the relative coordinates RC^l_{pred} and feature embedding F^l_e. In addition, we input the warped point cloud $P^l_{2,warp}$ (will be introduced in the following section) to GW-SVD.

For solving poses, by adding RC^l_{pred} to $P^l_{2,warp}$, a predicted target point cloud $P^l_{2,pred}$ is generated. We can get the pose through weights and point cloud correspondences.

In our application, we find dynamic elements in the autonomous scene (*e.g.*, moving objects), occluded regions, and outlier predictions often result in anomalous distance distributions. To address this issue, we introduce a geometric consistency check (GEO) in our module. For the input point cloud correspondences $P^l_{2,warp}$ and $P^l_{2,pred}$, we first calculate the distance matrices for each. These matrices represent the distances from each point in the point cloud to all other points. Next, we compute the cross-distance matrix, which represents the distance from each point in P^l_2 to its corresponding predicted point in $P^l_{2,pred}$. Following prior work (Bai et al. 2021), geometric consistency is defined as a binary matrix, where the value is 0 if the distance is below a predefined threshold d, and 1 otherwise. Finally, confidence is represented by the leading eigenvector. Our GEO performs statistical analysis on the relative coordinate distributions, assigning lower weights to outliers that deviate from the normal range, thereby generating geometrically-consistent confidence weights for each point.

The final confidence weight W^l is obtained by multiplying the feature-based and geometry-based weights. We then perform a weighted alignment between $P^l_{2,warp}$ and $P^l_{2,pred}$, followed by a differentiable SVD to estimate the pose incre-

ment at layer l, denoted as ΔR^l and Δt^l, as follows:

$$W^l = \text{Normalize}\left(\text{MLP}(F_e^l) \cdot \text{GEO}\left(P_{2,\text{warp}}^l, RC_{pred}^l\right)\right), \tag{7}$$

$$\Delta R^l, \Delta t^l = \text{SVD}\left(\overline{P}_{2,warp}^l \cdot \text{diag}(W^l) \cdot \left(\overline{P}_{2,pred}^l\right)^\top\right), \tag{8}$$

where $\overline{P}_{2,warp}^l$ represents the $P_{2,warp}^l$ that has been weighted by W^l and decentralized. $P_{2,pred}^l$ undergoes the same processing. Finally, the pose is obtained through SVD. The estimated pose increment ΔR^l, Δt^l is used to warp the point cloud, which is then forwarded to the next stage for further refinement.

Pose Refinement and Relative Coordinate Adjustment. In this module, the pose predictions ΔR^{l+1} and Δt^{l+1} from the previous GW-SVD layer are first obtained. These estimates are then used to update the coordinates of P_2 at all i ($i \leq l+1$) scale. Specifically, the updated $P_{2,warp}^i$ is computed by applying ΔR^{l+1} and Δt^{l+1} to the old $P_{2,warp}^i$ from the previous. The resulting point cloud is then fed into the next refinement layer l for further relative coordinate prediction and pose refinement.

$$P_{2,warp}^i = \Delta R^{l+1} \cdot P_{2,warp}^i + \Delta t^{l+1}, \qquad l < 3. \tag{9}$$

Note that $P_{2,warp}^3$ is equal to P_2^3.

In the next layer, the ground-truth relative coordinates RC_{gt}^l are updated as follows:

$$RC_{gt}^l = P_{gt}^l - P_{2,warp}^l, \qquad l < 3. \tag{10}$$

Additionally, the Level-l pose (R^l, t^l) can be obtained as:

$$R^l = \begin{cases} \Delta R^l, & l = 3 \\ \Delta R^l \cdot R^{l+1}, & l < 3 \end{cases} \tag{11}$$

$$t^l = \begin{cases} \Delta t^l, & l = 3 \\ \Delta R^l \cdot t^{l+1} + \Delta t^l, & l < 3 \end{cases} \tag{12}$$

Through this iterative process, the network progressively aligns the point clouds and refines the relative pose, yielding the final prediction at the last level.

Loss Function

The network outputs the rotation matrix R and the translation vector t from 4 different levels of the point cloud, which are used to compute the supervision loss. To mitigate the impact of scale and unit differences between the t and R, we introduce two learnable parameters s_x and s_r, following previous work (Li et al. 2019). The pose loss at the l-th layer is:

$$\begin{aligned} \ell_{rt}^l = &\left\|t_{pred} - t_{gt}\right\|_2 \exp(-s_x) + s_x \\ &+ \left\|\frac{R_{pred}^T R_{gt} - \mathbf{I}}{\sqrt{3}}\right\|_F \exp(-s_r) + s_r, \end{aligned} \tag{13}$$

where $\|\cdot\|_2$ represents the L_2 norm, and $\|\cdot\|_F$ denotes the Frobenius norm. t_{gt} and R_{gt} represent the ground-truth translation vector and ground-truth rotation matrix. $\mathbf{I}$ denotes the identity matrix.

The total loss for the L-layer supervision is:

$$\ell = \sum_{l=0}^{L} \alpha^l \ell_{rt}^l + \sum_{l=0}^{L-1} \beta^l \ell_{res}^l, \tag{14}$$

where L represents the number of iterative refinement layers in the network, and α^l denotes the fixed weight for the l-th layer. ℓ_{res}^l is the diffusion loss, which is defined in Eq. 6. Both diffusion and pose losses are jointly optimized.

Experiments

Dataset

KITTI/KITTI-360 The KITTI odometry dataset (Geiger et al. 2013) contains 22 independent sequences collected in the Karlsruhe region of Germany, covering diverse road types such as rural roads, urban areas, and highways, with dynamic elements including pedestrians, cyclists, and various vehicles. We use point cloud data from the Velodyne HDL-64E LiDAR with 64 beams and a 10 Hz sampling rate.

KITTI-360 (Liao, Xie, and Geiger 2022) represents an extension of the benchmark KITTI. In this study, we evaluate the performance using the official SLAM test trajectories provided with the dataset.

Ford Campus The Ford Campus Vision and LiDAR dataset (Pandey, McBride, and Eustice 2011) consists time-synchronized 2D panoramic image, 3D LiDAR point clouds and IMU/GPS data. Similar to KITTI, the LiDAR data is collected using a Velodyne HDL-64E sensor mounted on the vehicle's roof. The dataset contains two loop closure sequences collected in different urban environments. It contains more moving vehicles than the KITTI, resulting in more complex and dynamic scenes.

Oxford Radar RobotCar Oxford Radar RobotCar (Barnes et al. 2020) is an urban scene localization benchmark comprising over 32 traversals of a center Oxford route (10 km, 200 km^2), captured under varying weather, traffic, and lighting conditions. The point cloud is collected by dual Velodyne HDL-32E LiDAR. Ground truth pose is generated by the interpolations of INS. We utilized data collected by the left-mounted LiDAR and selected two sequences recorded on 2019-01-10(3.08 km) and 2019-01-15(4.00 km) as test sets.

Experiment Settings

Our network is implemented using PyTorch, with all training and testing conducted on an NVIDIA GeForce RTX 4090 GPU. The point cloud is first downsampled using a voxel grid filter with a voxel size of 0.2 m, then a distance-based sampling strategy is used to select points closer to the origin, downsampling each frame to 8,192 points. We employ the AdamW optimizer with β_1=0.9 and β_2 = 0.999, and an initial learning rate of 0.001, which decays by a factor of 0.8 every 9 epochs. We leverage DDPM (Ho, Jain, and Abbeel 2020) with the total timestamp k as 300. The threshold d of GW-SVD is set to 0.05 m. For loss function, the trainable parameters s_x and s_r in Eq. 13 are initialized to 0.0 and -2.5, respectively. The weighting coefficients used in Eq. 14

Method	07		08		09		10		Mean on 07-10	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
ICP-po2po	5.17	3.35	10.04	4.93	6.93	2.89	8.91	4.74	7.76	3.98
ICP-po2pl	1.55	1.42	4.42	2.14	3.95	1.71	6.13	2.60	4.01	1.97
GICP (Segal, Haehnel, and Thrun 2009)	0.64	0.45	1.58	0.75	1.97	0.77	1.31	0.62	1.38	0.65
CLS (Velas, Spanel, and Herout 2016)	1.04	0.73	2.14	1.05	1.95	0.92	3.46	1.28	2.15	1.00
A-LOAM w/o map (Zhang, Singh et al. 2014)	2.89	1.80	4.82	2.08	5.76	1.85	3.61	1.76	3.89	1.64
Full A-LOAM (Zhang, Singh et al. 2014)	0.69	0.50	1.18	0.44	1.20	0.48	1.51	0.57	1.15	0.50
LO-Net (Li et al. 2019)	1.70	0.89	2.12	0.77	1.37	0.58	1.80	0.93	1.75	0.79
SelfVoxeLO (Xu et al. 2021)	2.51	1.15	2.65	1.00	2.86	1.17	3.22	1.26	2.81	1.15
PWCLO-Net (Wang et al. 2021b)	0.60	0.44	1.26	0.55	0.79	0.35	1.69	0.62	1.09	0.49
RSLO (Xu et al. 2022a)	2.37	1.15	2.14	0.92	2.61	1.05	2.33	0.94	2.36	1.02
EffLO (Wang et al. 2022)	**0.46**	**0.38**	1.14	**0.41**	0.78	**0.33**	0.80	**0.46**	0.80	**0.40**
DELO (Ali et al. 2023)	0.58	0.41	1.36	0.64	1.23	0.57	1.53	0.90	1.18	0.63
TransLO (Liu et al. 2023a)	0.55	0.43	1.29	0.50	0.95	0.46	1.18	0.61	0.99	0.50
DSLO (Zhang et al. 2024a)	0.58	0.41	1.16	0.51	0.72	**0.33**	1.29	0.49	0.94	0.44
Ours	0.59	0.57	**0.90**	0.48	**0.68**	0.34	**0.70**	0.48	**0.72**	0.46

Table 1: The LiDAR odometry experiment results on KITTI odometry dataset. t_{rel}, r_{rel} indicate the average translational RMSE (%) and rotational RMSE (°/100m) respectively on all possible subsequences in the length of 100, 200, ..., 800m. The best result for each sequence is **bold**, and the second best is underlined.

Method	**360-test1**		**360-test2**		**Ford-1**		**Ford-2**		**Oxford-1**		**Oxford-2**	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Full A-LOAM (Zhang, Singh et al. 2014)	**1.59**	**0.63**	**1.98**	**0.71**	**1.88**	**0.50**	**2.05**	**0.56**	**4.95**	**1.44**	**4.80**	**1.45**
ICP-po2po	7.55	3.44	7.26	3.12	8.20	2.64	16.23	2.84	24.99	7.87	22.59	6.90
ICP-po2pl	2.12	1.19	3.40	2.75	3.35	1.65	5.68	1.96	12.09	5.73	12.85	5.33
GICP (Segal, Haehnel, and Thrun 2009)	1.49	**0.90**	3.53	1.70	3.07	**1.17**	5.11	1.47	**6.15**	3.74	**7.04**	5.98
A-LOAM w/o map (Zhang, Singh et al. 2014)	4.89	2.78	5.14	2.52	4.17	2.00	4.72	1.65	19.38	7.99	17.98	7.69
EffLO (Wang et al. 2022)	**1.38**	1.08	2.71	1.64	6.95	3.88	10.90	6.30	39.04	20.07	32.89	17.01
TransLO (Liu et al. 2023a)	1.78	1.14	4.08	2.14	14.05	5.79	18.33	6.21	50.86	26.24	72.85	30.34
Ours	1.66	1.12	**2.60**	**1.34**	**2.63**	1.31	**3.71**	**1.22**	7.07	**3.04**	7.59	**3.66**

Table 2: Evaluation Results on other datasets. The best performance is **bold**.

are set to [1.6, 0.8, 0.4, 0.2], and the batch size is fixed at 4 during training.

Comparison with State-of-the-arts

Evaluation on KITTI We use sequences 00–06 from the KITTI dataset for training and evaluate on sequences 07–10. The quantitative results of different methods on the test set are summarized in Tab. 1.

Compared to recent learning-based LiDAR odometry methods such as LO-Net (Li et al. 2019), PWCLONet (Wang et al. 2021b), RSLO (Xu et al. 2022b), EfficientLO (Wang et al. 2022), DELO (Ali et al. 2023), TransLO (Liu et al. 2023a), and DSLO (Zhang et al. 2024a), our proposed method achieves the lowest average translation error (0.72%) among all learning-based methods while maintaining a competitively low average rotation error (0.46°/100m). Fig. 4 presents the comparative average translation and rotation errors on the all KITTI sequence.

Generalization to Other Datasets To validate the generalization capability of the model, we evaluate it on other datasets (Liao, Xie, and Geiger 2022; Pandey, McBride, and Eustice 2011; Barnes et al. 2020). For a fair comparison, all learning-based models are trained on KITTI sequences 00–06 and evaluated without any fine-tuning. The results are summarized in Tab. 2.

	Methods	Mean on 07-10	
		t_{rel}	r_{rel}
Regression Target	Pose	0.89	**0.43**
	RC	**0.72**	0.46
Pose Solver	MLP	0.82	**0.39**
	Vanilla-SVD	1.07	0.58
	Mask-SVD	1.22	0.77
	GW-SVD	**0.72**	0.46
Hierarchical Refinement	Cost-volume	0.86	0.53
	Diffusion	**0.72**	**0.46**
Hierarchical Levels	T3-layer	4.42	2.16
	T2-layer	2.24	1.14
	T1-layer	1.04	0.56
	T0-layer	**0.72**	**0.46**

Table 3: Ablation results on the KITTI odometry dataset.

On the KITTI-360 dataset, our method achieves localization accuracy comparable to state-of-the-art learning-based approaches TransLO (Liu et al. 2023a) and EffLO (Wang et al. 2022). This is expected, as KITTI-360 shares the same sensor configuration and similar environments with KITTI, allowing stable cross-scene generalization.

Dataset	Baseline		Ours	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}
KITTI	0.89	0.43	0.72	0.46
KITTI-360	1.96	1.18	2.13	1.23
Ford	7.02	2.80	3.17	1.27
Oxford	30.24	14.92	7.33	3.35

Table 4: Ablation study on KITTI, KITTI-360, Ford, and Oxford datasets.

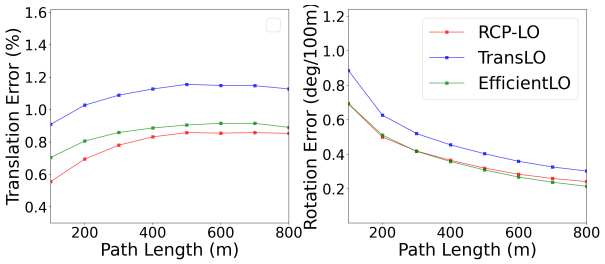

Figure 4: Average translation and rotation error on the KITTI sequences 00-10 on all possible subsequences with the length of 100, 200,..., 800 m.

On the Ford Campus dataset, the compared learning-based methods show significant performance degradation. In contrast, our approach attains 2.63% translation and 1.31°/100m rotation error on Ford-1, and 3.71% / 1.22°/100m on Ford-2, reaching the performance level of traditional methods. The 3D trajectory in Fig. 5 further illustrates the robustness of our approach in complex and previously unseen environments.

On the Oxford RobotCar dataset, all methods experience reduced accuracy due to the extreme sparsity of the LiDAR scans. While the learning-based baselines fail to operate reliably, our method maintains accuracy comparable to GICP (Segal, Haehnel, and Thrun 2009). RCP-LO predicts correspondences and confidence weights analogously to GICP's correspondence search and covariance weighting, we believe this geometry-driven design to its strong cross-dataset generalization, as it relies on local geometric alignment rather than dataset-specific appearance cues.

Ablation Study

To evaluate the effectiveness of our model components, we perform a comprehensive ablation study on the KITTI dataset. The results are summarized in Tab. 3.

Pose Solver. We further evaluate the impact of using an MLP for direct relative pose regression and different SVD-based pose estimation strategies. compared with MLP-based pose regression, unweighted Vanilla-SVD and feature-similarity-based Mask-SVD,which highlight the robustness of GW-SVD in suppressing the adverse impact of unreliable point correspondences by assigning them lower confidence weights.

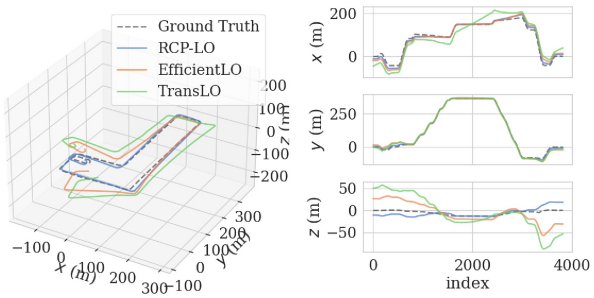

Figure 5: 3D trajectory results of Ford-1.

Hierarchical Refinement. To assess the contribution of the diffusion module, we replaced it with the attentive cost-volume module proposed in (Wang et al. 2021a) for relative coordinate prediction. The experimental results reveal a notable performance degradation, which substantiates the effectiveness of our diffusion strategy in generating accurate relative coordinates for pose estimation.

Hierarchical Levels. We report the localization accuracy at each refinement layer to validate the effectiveness of the hierarchical pose refinement design. The results demonstrate that the hierarchical refinement strategy significantly improves pose estimation accuracy by progressively refining from sparse to dense point clouds, effectively leveraging the increasingly detailed geometric information available in denser point clouds.

Generalization about Baseline. The baseline adopts the same backbone as RCP-LO but replaces the coordinate-prediction branch with a cost-volume and MLP-based pose regression head, similar to (Wang et al. 2021b). Extensive experiments across diverse scenarios and LiDAR beam configurations (Tab. 4) demonstrate our framework's superior generalization over the baseline.

Runtime Analysis

The real-time capability of pose inference is critical for odometry tasks. The inference time for estimating the relative pose between two consecutive point clouds is 83.9ms, which meets real-time requirements and demonstrates a strong potential for practical deployment.

Conclusion

We propose RCP-LO, a novel Deep LiDAR odometry framework that reformulates relative pose estimation as relative coordinate prediction, preserving the geometric structure of point clouds and enabling robust and accurate pose estimation. We introduce a diffusion model to predict accurate relative coordinates from point clouds. Furthermore, we design a GW-SVD module that allows for efficient pose estimation in a single forward pass. Experiments demonstrate that RCP-LO, trained solely on the KITTI dataset, achieves state-of-the-art performance and exhibits notable generalization to unseen datasets.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (No. 62501502).

References

Ali, S. A.; Aouada, D.; Reis, G.; and Stricker, D. 2023. Delo: deep evidential lidar odometry using partial optimal transport. In *ICCV*, 4517–4526.

Arun, K. S.; Huang, T. S.; and Blostein, S. D. 1987. Least-squares fitting of two 3-D point sets. *IEEE TPAMI*, (5): 698–700.

Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; and Tai, C.-L. 2021. Pointdsc: Robust point cloud registration using deep spatial consistency. In *CVPR*, 15859–15869.

Barnes, D.; Gadd, M.; Murcutt, P.; Newman, P.; and Posner, I. 2020. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *ICRA*, 6433–6438.

Chetverikov, D.; Svirko, D.; Stepanov, D.; and Krsek, P. 2002. The trimmed iterative closest point algorithm. In *ICCV*, volume 3, 545–548.

Dellenbach, P.; Deschaud, J.-E.; Jacquet, B.; and Goulette, F. 2022. Ct-icp: Real-time elastic lidar odometry with loop closure. In *ICRA*, 5580–5586.

Fischler, M. A.; and Bolles, R. C. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 15: 381–395.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *IJRR*, 32(11): 1231–1237.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851.

Hu, H.; Qiao, Z.; Cheng, M.; Liu, Z.; and Wang, H. 2020. Dasgil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE TIP*, 30: 1342–1353.

Koide, K.; Yokozuka, M.; Oishi, S.; and Banno, A. 2021. Voxelized GICP for fast and accurate 3D point cloud registration. In *ICRA*, 11054–11059.

Li, Q.; Chen, S.; Wang, C.; Li, X.; Wen, C.; Cheng, M.; and Li, J. 2019. Lo-net: Deep real-time lidar odometry. In *CVPR*, 8473–8482.

Li, W.; Yu, S.; Wang, C.; Hu, G.; Shen, S.; and Wen, C. 2023. SGLoc: Scene geometry encoding for outdoor LiDAR localization. In *CVPR*, 9286–9295.

Li, X.; Wang, S.; Zhao, Y.; Verbeek, J.; and Kannala, J. 2020. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 11983–11992.

Li, Z.; and Wang, N. 2020. Dmlo: Deep matching lidar odometry. In *IROS*, 6010–6017.

Liao, Y.; Xie, J.; and Geiger, A. 2022. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI*, 45(3): 3292–3310.

Liu, J.; Wang, G.; Jiang, C.; Liu, Z.; and Wang, H. 2023a. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *AAAI*, volume 37, 1683–1691.

Liu, J.; Wang, G.; Liu, Z.; Jiang, C.; Pollefeys, M.; and Wang, H. 2023b. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. In *CVPR*, 8451–8460.

Liu, J.; Wang, G.; Ye, W.; Jiang, C.; Han, J.; Liu, Z.; Zhang, G.; Du, D.; and Wang, H. 2024. Difflow3d: toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In *CVPR*, 15109–15119.

Liu, J.; Zhuo, D.; Feng, Z.; Zhu, S.; Peng, C.; Liu, Z.; and Wang, H. 2025. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment. In *ECCV*, 475–493.

Liu, X.; Qi, C. R.; and Guibas, L. J. 2019. Flownet3d: Learning scene flow in 3d point clouds. In *CVPR*, 529–537.

Liu, Z.; Zhou, S.; Suo, C.; Yin, P.; Chen, W.; Wang, H.; Li, H.; and Liu, Y.-H. 2019. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, 2831–2840.

Lu, F.; Chen, G.; Liu, Y.; Zhang, L.; Qu, S.; Liu, S.; and Gu, R. 2021. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *ICCV*, 16014–16023.

Pandey, G.; McBride, J. R.; and Eustice, R. M. 2011. Ford campus vision and lidar data set. *IJRR*, 30(13): 1543–1552.

Park, K.; Patten, T.; and Vincze, M. 2019. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *ICCV*, 7668–7677.

Sattler, T.; Zhou, Q.; Pollefeys, M.; and Leal-Taixe, L. 2019. Understanding the Limitations of CNN-Based Absolute Camera Pose Regressionr. In *CVPR*, 3302–3312.

Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-icp. In *Robotics: science and systems*, volume 2, 435.

Shan, T.; and Englot, B. 2018. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IROS*, 4758–4765.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-Or, D.; and Bermano, A. H. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.

Velas, M.; Spanel, M.; and Herout, A. 2016. Collar line segments for fast odometry estimation from velodyne point clouds. In *ICRA*, 4486–4495.

Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J.; and Stachniss, C. 2023. Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *RAL*, 8(2): 1029–1036.

Wang, G.; Wu, X.; Jiang, S.; Liu, Z.; and Wang, H. 2022. Efficient 3d deep lidar odometry. *IEEE TPAMI*, 45(5): 5749–5765.

Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021a. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE TIP*, 30: 5168–5181.

Wang, G.; Wu, X.; Liu, Z.; and Wang, H. 2021b. Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization. In *CVPR*, 15910–15919.

Wang, H.; Wang, C.; Chen, C.-L.; and Xie, L. 2021c. F-loam: Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4390–4396.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024. Dust3r: Geometric 3d vision made easy. In *CVPR*, 20697–20709.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 9621–9630.

Wu, X.; Zhao, H.; Li, S.; Cao, Y.; and Zha, H. 2022. Sc-wls: Towards interpretable feed-forward camera re-localization. In *ECCV*, 585–601.

Xu, Y.; Huang, Z.; Lin, K.-Y.; Zhu, X.; Shi, J.; Bao, H.; Zhang, G.; and Li, H. 2021. Selfvoxelo: Self-supervised lidar odometry with voxel-based deep neural networks. In *CoRL*, 115–125.

Xu, Y.; Lin, J.; Shi, J.; Zhang, G.; Wang, X.; and Li, H. 2022a. Robust self-supervised lidar odometry via representative structure discovery and 3d inherent error modeling. *RAL*, 7(2): 1651–1658.

Xu, Y.; Lin, J.; Shi, J.; Zhang, G.; Wang, X.; and Li, H. 2022b. Robust self-supervised lidar odometry via representative structure discovery and 3d inherent error modeling. *IEEE RAL*, 7(2): 1651–1658.

Yang, B.; Li, Z.; Li, W.; Cai, Z.; Wen, C.; Zang, Y.; Muller, M.; and Wang, C. 2024. LiSA: LiDAR Localization with Semantic Awareness. In *CVPR*, 15271–15280.

Zhang, H.; Wang, G.; Wu, X.; Xu, C.; Ding, M.; Tomizuka, M.; Zhan, W.; and Wang, H. 2024a. DSLO: Deep Sequence LiDAR Odometry Based on Inconsistent Spatio-temporal Propagation. In *IROS*, 10672–10677.

Zhang, J.; Singh, S.; et al. 2014. Lidar odometry and mapping in real-time. In *RSS*, volume 2, 1–9.

Zhang, J. Y.; Lin, A.; Kumar, M.; Yang, T.-H.; Ramanan, D.; and Tulsiani, S. 2024b. Cameras as rays: Pose estimation via ray diffusion. In *ICLR*.

Zheng, C.; Lyu, Y.; Li, M.; and Zhang, Z. 2020. Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation. In *ACMMM*, 2391–2399.