DiffLO: Semantic-Aware LiDAR Odometry with Diffusion-Based Refinement (SupplementaryMaterial)

Yongshu Huang[1*] Chen Liu[1*] Minghang Zhu[1] Sheng Ao[1†] Chenglu Wen[1] Cheng Wang[1†]
[1] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, China.
[2] Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, China.

In this supplementary, we first provide additional details on the datasets and data preprocessing (Sec. 1). We then describe the implementation details of the network (Sec. 2). Next, we present further ablation experiments (Sec. 3). Finally, we provide additional visualizations of the KITTI odometry datasets (Sec. 4).

1. Datasets and Data Pre-processing

1.1. Dataset

The KITTI odometry dataset[1] consists of 22 independent sequences, specifically designed for evaluating visual and LiDAR-based odometry and SLAM methods. In our approach, we focus exclusively on the Velodyne LiDAR point cloud data, utilizing the XYZ coordinates provided in the scans. Sequences 00-10, comprising 23,201 scans, are accompanied by ground truth poses (trajectories), enabling training and quantitative evaluation. However, ground truth data is not publicly available for sequences 11-21, which include an additional 20,351 scans. The dataset captures diverse driving environments, such as highways, residential areas, and campus roads, providing a rich and varied set of point cloud data for benchmarking LiDAR odometry tasks.

1.2. Data Preprocessing

We adopted a data preprocessing approach similar to that described in [4]. Specifically, we only use the coordinate information of LiDAR points. Since the ground truth poses are provided in the left camera coordinate system, both the training and evaluation processes are conducted within this coordinate system. To achieve this, the point clouds captured by the Velodyne LiDAR are transformed into the left camera coordinate system using the following equation:

$$P_{cam} = T_r P_{vel}, \tag{1}$$

where P_{cam} and P_{vel} represent the coordinates of the point cloud in the left camera coordinate system and the LiDAR coordinate system, respectively, and T_r is the calibration matrix of each sequence. In addition, to handle potential outliers in the point cloud, often appearing at the edges due to objects being far from the LiDAR sensor, we exclude LiDAR points located beyond a 30-meter by 30-meter bounding box surrounding the vehicle. Given that ground reflections in point cloud data often appear below 0.55 meters, these low-elevation artifacts are systematically eliminated. This preprocessing effectively improves the quality of the point cloud data while accelerating data loading and training.

1.3. Data Augmentation Parameters

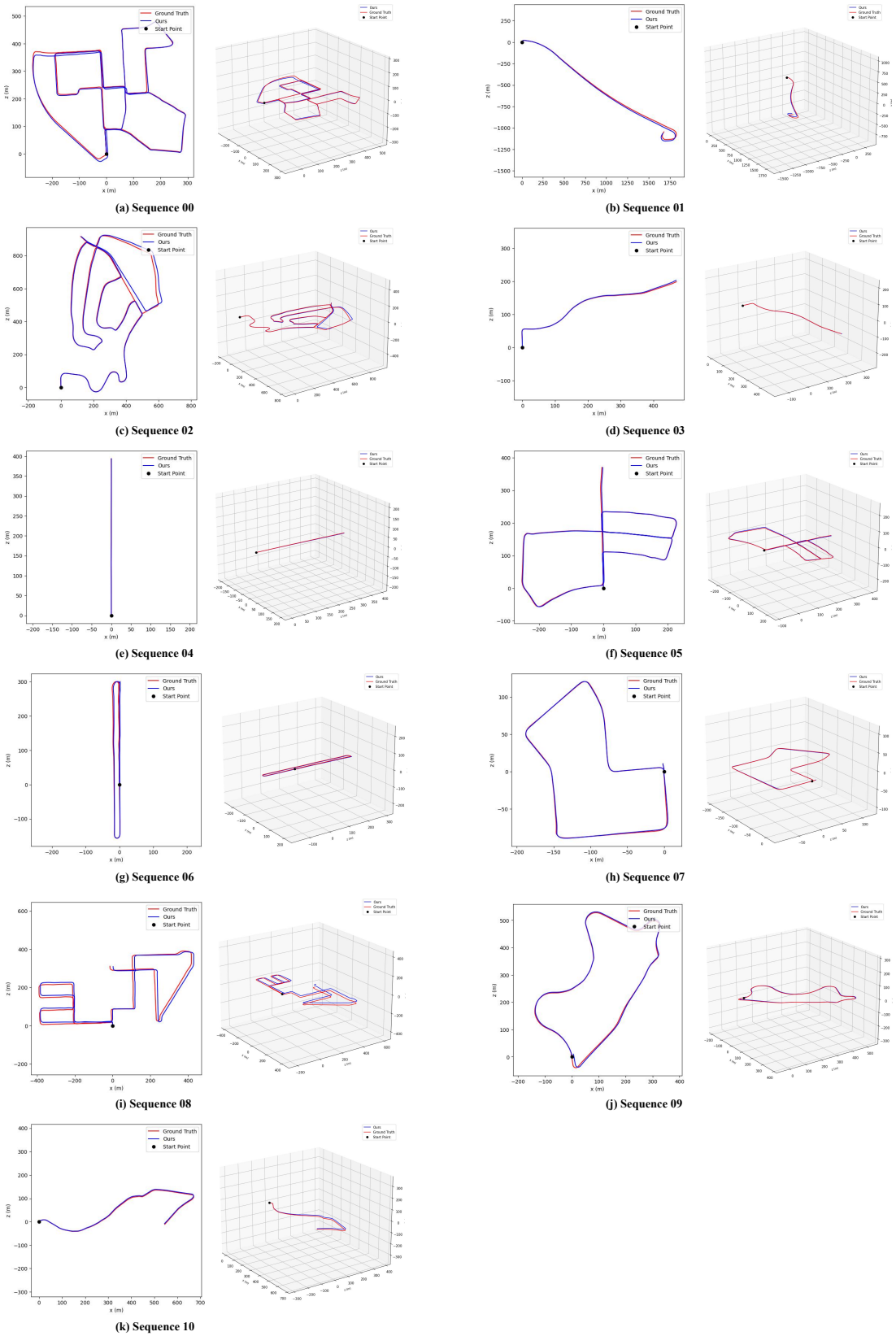
We adopted a data augmentation approach similar to that described in [4]. Specifically, we augment the training data by applying an augmentation matrix T_{aug}, which is composed of a rotation matrix R_{aug} and a translation vector t_{aug}.The yaw, pitch, and roll Euler angles for R_{aug} are sampled from Gaussian distributions centered at $0°$, with standard deviations of $0.01°$, $0.05°$ and $0.01°$, respectively. Similarly, the translation components X, Y, and Z of t_{aug} follow Gaussian distributions with standard deviations of $0.1m$, $0.1m$, and $0.5m$, respectively. To ensure consistency and avoid extreme perturbations, only values within twice the standard deviation of the mean are selected for augmentation. The generated transformation T_{aug} is then applied to the point cloud PC_1, resulting in the augmented point cloud $PC_{1,aug}$. This augmentation strategy effectively enhances the diversity of the training dataset while maintaining realistic variations.

2. Implementation Details

2.1. Training and Inference Process

Training. Our diffusion model adopts the same sampling and training strategy as described in DDIM [2]. Specifically, the training process is outlined in Algorithm 1. To begin, the intermediate pose residual T_t is sampled from the forward diffusion process by applying a predefined noise

*These authors contributed equally.
†Corresponding Author.



(a) Sequence 00 (b) Sequence 01

(c) Sequence 02 (d) Sequence 03

(e) Sequence 04 (f) Sequence 05

(g) Sequence 06 (h) Sequence 07

(i) Sequence 08 (j) Sequence 09

(k) Sequence 10

Figure 1. 3D and 2D trajectory results of our network on KITTI sequences 00-10, compared with ground truth.

Method	07†		08†		09†		10†		Mean on 07-10		Runtime(ms)
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	
w/o diffusion-based refinement layer	4.83	3.82	6.77	3.05	5.29	2.22	7.09	3.97	5.995	3.265	7.4
Predicted pose with one refinement layer	1.02	0.91	1.58	0.73	1.21	0.62	2.18	1.29	1.498	0.888	25.7
Predicted pose with two refinement layers	0.56	0.47	1.32	0.55	0.58	0.35	1.25	0.80	0.928	0.543	32.9
Ours (Predicted pose with three refinement layers)	**0.37**	**0.27**	**1.12**	**0.44**	**0.68**	**0.28**	**0.66**	**0.32**	**0.708**	**0.328**	76.9

Table 1. Ablation studies about the layer number of diffusion-based refinement on the KITTI odometry [1] dataset.

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}
Layer-0 Semantic Supervision	0.42	0.30	**1.12**	**0.44**	0.71	**0.28**	0.81	0.53	0.765	0.388
Ours (Layer-3 Semantic Supervision)	**0.37**	**0.27**	**1.12**	**0.44**	**0.68**	**0.28**	**0.66**	**0.32**	**0.708**	**0.328**

Table 2. Ablation Studies on Semantic Feature Supervision at Different Levels on the KITTI Odometry [1] Dataset

Algorithm 1 Training Process

Input: timestamp t, ground truth pose residual T_0, condition information C.

repeat
  $T_0 \sim q(T_0)$
  $t \sim$Uniform$(\{1, \ldots, K\})$
  $T_t = \sqrt{\overline{\alpha}_t} T_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathrm{I})$
  optimize loss: $\mathcal{L} = loss(T_0, \mathcal{M}_\theta(T_t, C, t))$
until converged

Algorithm 2 Sampling Process

Input: timestamp t, Gaussian noise T_t, condition information C.
Output: refined pose resiudal T_0

$T_k \sim \mathcal{N}(0, I)$
for $t = k, \ldots, 1$ **do**
  if $t > 1$: $z \sim \mathcal{N}(0, I)$
  else: $z = 0$
  $T_{t-1} = \frac{1}{\sqrt{\alpha_t}}(T_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\mathcal{M}_\theta(T_t, C, t)) + \sigma_t z$
end for
return T_0

schedule to the ground truth residual pose T_0. During training, the condition information C, the intermediate pose residual T_t, and the time embedding t serve as inputs to the denoising network $\mathcal{M}_\theta$. The objective of the optimization process is to minimize the divergence between the ground truth pose residual T_0 and the predicted pose residual $\mathcal{M}_\theta(T_t, C, t)$, effectively narrowing the distribution gap. Our network employs a multi-scale supervision mechanism that integrates supervision from the pose, pose residual, and semantic features. Furthermore, to emphasize different refinement layers in the training process, we assign loss weights of 0.2, 0.4, 0.8, and 1.6 across successive layers. All variables and notations are defined in detail in the main manuscript.

Sampling. We iteratively generate pose residuals starting from an initial Gaussian noise T_t, as described in Algorithm 2. Our proposed diffusion-based refinement module progressively refines the pose residuals in a coarse-to-fine manner. Each refinement layer follows the same sampling process outlined in Algorithm 2.

3. Additional Experiments

Refinement layer number. We first evaluate the impact of the number of diffusion-based refinement layers in our model. As shown in Tab. 1, increasing the number of coarse-to-fine refinement layers initially improves the accuracy of the predicted pose. However, as the number of layers continues to increase, the computational cost rises significantly. This observation suggests that there exists an optimal balance between refinement depth and efficiency.

Layer-3 Semantic Supervision vs Layer-0 Semantic Supervision We explored the impact of supervising semantic features at different levels. As shown in Tab. 2, the experimental results demonstrate that supervising semantic features at Layer 3 achieves better performance compared to lower-level feature supervision. This highlights the effectiveness of leveraging higher-level semantic representations for improved results.

The Semantic Module network We evaluated the performance of different semantic regression subnetwork within the semantic module, including a transformer-based network [3] (referred to as "trans-based"). As shown in Tab. 3. The results highlight the superiority of our semantic regression subnetwork compared to alternative designs.

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}
trans-based Semantic Module	0.52	0.40	1.22	0.49	0.59	0.29	0.93	0.35	0.815	0.383
Ours	**0.37**	**0.27**	**1.12**	**0.44**	**0.68**	**0.28**	**0.66**	**0.32**	**0.708**	**0.328**

Table 3. Ablation Studies on Variants of the Semantic Regression Subnetwork in the Semantic Module on the KITTI Odometry [1] Dataset

4. More Trajectory Results

We list all visualized trajectory results on sequences 00-10 of KITTI odometry dataset [1] with the ground truth in Fig. 1. The figure shows our odometry can track the trajectory of the ground truth fairly well.

References

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32 (11):1231–1237, 2013. 1, 3, 4

[2] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[3] A Vaswani. Attention is all you need. *NeurIPS*, 2017. 3

[4] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Pwclo-net: Deep lidar odometry in 3d point clouds using hierarchical embedding mask optimization. In *CVPR*, pages 15910–15919, 2021. 1