DiffLO: Semantic-Aware LiDAR Odometry with Diffusion-Based Refinement

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

Abstract

LiDAR odometry is a critical module in autonomous driving systems, responsible for accurate localization by estimating the relative pose transformation between consecutive point cloud frames. However, existing studies frequently encounter challenges with unreliable pose estimation, due to the lack of in-depth understanding of scenario and the presence of noise interference. To address this challenge, we propose DiffLO, a semantic-aware LiDAR odometry network with diffusion-based refinement. To mitigate the impact of challenging cases such as dynamic, repetitive patterns, and low textures, we introduce a semantic distillation method that integrates semantic information into the odometry task. This allows the network to gain a semantic understanding of the scene, enabling it to focus more on the objects that are beneficial for pose estimation. Additionally, to enhance the robustness, we propose a diffusion-based refinement method. This method uses pose-related features as conditional constraints for generative diversity, iteratively refining the pose estimation to achieve greater accuracy. Comparative experiments on the KITTI odometry dataset demonstrate that the proposed method achieves state-of-the-art performance among existing learning-based approaches. Furthermore, the proposed DiffLO method outperforms the classic A-LOAM on most evaluation sequences. The code will be released at https://github.com/HyTree7/DiffLO.

1. Introduction

LiDAR odometry (LO) is a well-studied problem in computer vision with numerous applications in Simultaneous Localization and Mapping (SLAM) [58], autonomous driving [54], and robotics [45]. The primary objective is to estimate the relative transformation between two frames based on the source and target point clouds.

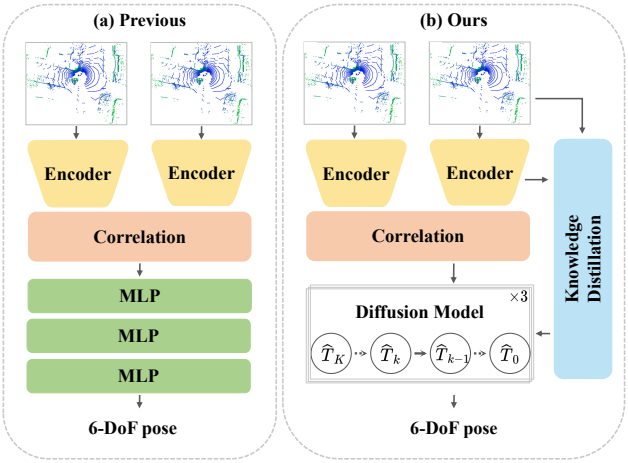


Figure 1. (a) Predicts a coarse initial pose and applies iterative MLP-based refinement to refine the initial estimate. (b) Predicts a coarse initial pose and applies iterative Diffusion-based refinement, incorporating knowledge distillation to enable semantic understanding. Here, $\hat{T}_k$ represents the noisy pose of the k^{th} denoising step.

Most classic geometry-based methods, such as Iterative Closest Point (ICP) [4], its variants [31] and others [35], work by iteratively minimizing the distance between corresponding points in two point clouds to achieve accurate alignment. However, they often struggle in sparse point cloud environments, complex dynamic scenes, or when the quality of the point cloud degrades [13, 27, 42]. This limitation arises due to their reliance on precise geometric correspondences.

Deep learning-based methods have garnered significant attention in recent years. These methods learn feature representations from large-scale data to infer relative transformation between frames in an end-to-end manner, providing better scalability and the potential for multi-task learning. However, the accuracy of deep learning methods still lags behind traditional odometry techniques.

We identify the following challenges faced by Deep Learning-based odometry methods: (1) To mitigate the effects of challenging cases such as dynamic objects and oc-

*These authors contributed equally.
†Corresponding Author.

clusions, LO-Net [21] introduces a mask prediction network learning the compensation for dynamic objects to remove dynamic objects. Methods such as PWCLO-Net [42], TransLO [23], and EfficientLO [44] introduce a hierarchical embedding mask and apply pose feature weighting to filter dynamic object. However, these approach lacks a comprehensive understanding of object structures, leading to inaccuracies and resulting in cumulative errors; (2) Furthermore, these methods [18, 23, 24, 26, 42, 44] are deterministic prediction tasks, which limits their ability to adapt to newly emerging changes or complexities in the environment. Additionally, they can not provide information about prediction uncertainties.

Semantic segmentation network contains rich geometric information and intricate details about the scene. This insight motivates us to integrate semantic information into the odometry network, thereby enhancing its ability to filter out dynamic objects and manage occlusions effectively. To achieve this, as shown in Fig. 1, we propose a knowledge distillation approach that transfers semantic knowledge from a segmentation model to the odometry network. After training, the knowledge distillation component and segmentation model are droped to avoid extra computation or network parameter. By leveraging the contextual knowledge provided by semantic segmentation, our approach aims to improve the overall accuracy and robustness of odometry in complex environments.

Diffusion models [11] are characterized by their robustness, and have achieved significant success across various tasks [8, 15, 22, 25]. These strengths have inspired us to explore their potential application to the LiDAR odometry task. In this paper, we propose reformulating the odometry task as a diffusion process. As shown in Fig. 1, we transition from a traditional MLP-based refinement approach to a diffusion-based refinement, shifting the task from a deterministic prediction framework to a probabilistic generative model. Specifically, Our approach consists of a forward process that progressively adds noise to the residual pose of the ground truth, followed by a reverse denoising process to refine the noisy pose estimate. We utilize coarse pose embedding, cost volume, semantic feature and geometry embedding as conditions to restrict generation diversity.

In summary, our contributions are as follows:

- To enhance the network's ability to focus on objects critical for pose estimation in the scene, we incorporate semantic awareness into odometry by distilling knowledge from a semantic segmentation model.
- We propose a novel diffusion-based refinement pipeline and design strong conditional guidance by combining semantic embedding, coarse pose embeddings, geometry embedding and cross-frame cost volume to control generation diversity. To the best of our knowledge, this is the first work to leverage diffusion probabilistic model in the LiDAR odometry task.
- Extensive experiments on the KITTI odometry dataset [7] demonstrate that our method outperforms all existing learning-based odometry methods. On most sequences, our performance even surpasses A-LOAM with mapping.

2. Relate Work

2.1. Deep LiDAR Odometry

The field of visual depth odometry has advanced rapidly. However, research on LiDAR-based depth odometry remains challenging due to the inherent irregularity and complexity of point clouds. LO-Net [21] proposed an end-to-end deep odometry network that enhances robustness by learning the compensation for dynamic objects through a mask prediction network. PWCLO [42] first introduces a coarse-to-fine structure to LiDAR odometry, implementing a hierarchical refinement approach for estimated poses. LodoNet [52] projects 3D point clouds onto 2D spherical depth images to extract and match keypoints for odometry localization. EfficientLO [44] introduces a projection-aware representation of 3D point clouds, facilitating efficient learning for large-scale datasets. TransLO [23] develops a cross-frame transformer module that associates consecutive frames, thereby regressing relative pose estimations. DELO [1] innovatively employs partial optimal transportation of LiDAR feature descriptors to achieve robust localization estimation.

2.2. LiDAR-based semantic segmentation

LiDAR-based semantic segmentation aims to assign a semantic label to each point in the input LiDAR scan. These methods can generally be categorized into several approaches: point-based [14, 32, 33, 39], voxel-based [9, 57], projection-based [17, 29, 46], and hybrid methods that combine these techniques [37, 49]. These approaches consistently expand the receptive field by integrating fine-grained point features with coarse-grained contextual information,which enhances the network's ability to understand the scene structure and the unique characteristics of each point. Semantic information has also proven to be highly valuable in traditional non-deep learning methods, such as [3, 5, 19, 20, 56]. By leveraging semantic labels, these systems can enhance feature matching and improve robustness against challenging conditions like dynamic objects and occlusions.

Inspired by these traditional methods, we explore integrating semantics into learning-based LiDAR odometry to address the challenges posed by dynamic objects, repetitive patterns, and low-texture environments.

2.3. 3D Diffusion Model

Diffusion models have garnered significant attention and research in a variety of 3D point cloud processing tasks. For instance, Luo et al. [28] introduced a probabilistic model for point cloud generation, transforming the point cloud noise distribution into the desired point cloud shape distribution. Zhou et al. [55] combined a denoising diffusion model with a mixed point-voxel representation of 3D shapes for unconditional shape generation and conditional multimodal shape completion. Jiang et al. [15] proposed a point cloud registration framework based on an SE(3) diffusion model, framing the 3D registration task as a denoising diffusion process that iteratively refines the pose of the source point cloud to achieve precise alignment with the target point cloud. Diffflow3D [25] presented an uncertainty-aware scene flow estimation network leveraging a diffusion probabilistic model, where iterative diffusion modules refine initial estimates to enhance the robustness of scene flow estimation. DiffLoc [22] formulates LiDAR localization as a conditional generation of poses, employing a diffusion model conditioned on geometric robust features and incorporating an iterative denoising process into APR to achieve accurate LiDAR localization.

Inspired by the successful applications of diffusion models in tasks such as registration, scene flow estimation, and localization, we investigate the integration of diffusion models to improve the accuracy of LiDAR odometry.

3. Method

Given two consecutive point cloud frames $PC_1 \in \mathbb{R}^{N\times 3}$ and $PC_2 \in \mathbb{R}^{M\times 3}$, the goal of the LiDAR odometry task is to estimate the relative transformation between the two consecutive frames of point clouds. To solve this task, we propose DiffLO, a semantic-aware odometry network with diffusion-based refinement. DiffLO first hierarchically extracts point features (Sec. 3.1). Subsequently, we introduce a Semantic Perception Module (Sec. 3.2), which enables the odometry network to gain semantic understanding. Then, the initial pose is generated (Sec. 3.3) in the coarsest layer. We then utilize diffusion model with condition signals to refine the initial pose by predicting pose residuals from coarse to fine (Sec. 3.4). Lastly, the loss function will be presented in (Sec. 3.5).

3.1. Multi-scale Feature Extraction

We utilize PointConv [47] for hierarchical feature extraction. At each level l, dense input points and their features are first subsampled using furthest point sampling to create a sparse point set. Then, for each subsampled sparse point, k-nearest neighbor groups local dense points to form a local region for feature extraction. Next, a PointConv [47] layer aggregates features from the grouped local points, producing a local feature for each sparse point. This process results in an L-level pyramid of point features.

3.2. Semantic Perception Module

To guide the network's attention towards objects in the scene that are beneficial for pose estimation, such as buildings and pole-like structures. We introduce Semantic Perception Module (SPM). SPM consists of three components: Semantic Module, Semantic Segmentation Modle and Knowledge Distillation. We introduce a semantic segmentation model as the teacher network. A sub-network (the Semantic Module) within the odometry model is designated as the student model.

During the training stage, three modules collaborate to transfer semantic knowledge into the odometry model. Firstly, the Semantic Module takes the point cloud features F in coarse layer as its input, the output can be expressed as follows:

$$F_{stu} = MLP(F) + MLP(F) \otimes \sigma(MLP(F)), \quad (1)$$

where $\otimes$ indicates element-wise multiplication. $\sigma(\cdot)$ denotes the softmax operation along the feature dimension.

Secondly, a pretrained 3D semantic segmentation model [16], with its network parameters frozen, is utilized here. We extract the per-point features from the final layer, denoted as $F_{seg} \in \mathbb{R}^{N\times d}$. These features are subsequently downsampled to $F_{tea} \in \mathbb{R}^{N'\times d}$ to ensure compatibility in dimension with F_{stu}. And we adjust the output dimension of the semantic regressor branch, ensuring that F_{tea} and F_{stu} share the same feature dimension d for each point.

Then, we utilize knowledge distillation [10] to transfer semantic knowledge from the segmentation module to the odometry model.This is achieved by optimizing the following distillation loss:

$$\mathcal{L}_{kd} = \delta(F_{stu}, F_{tea}), \quad (2)$$

where $\delta(\cdot,\cdot)$ measures the difference between the feature representations from the teacher network F_{tea} and the student network F_{stu}.

Finally, the semantic feature F_{stu} is propagated to the following level for a denser optimization and simultaneously serves as a condition signal for diffusion-based refinement in Sec. 3.4, via an upsampling layer [6, 48].

During the inference stage, only the Semantic Module is retained to avoid additional computational overhead. The output F_{stu} from the Semantic Module is still propagated to subsequent layers, serving as a conditional signal.

By embedding semantic understanding into the odometry process, the model is better able to focus on objects in the scene that are beneficial for pose estimation. Meanwhile, the close integration of knowledge distillation improves both the efficiency and effectiveness of the overall system.

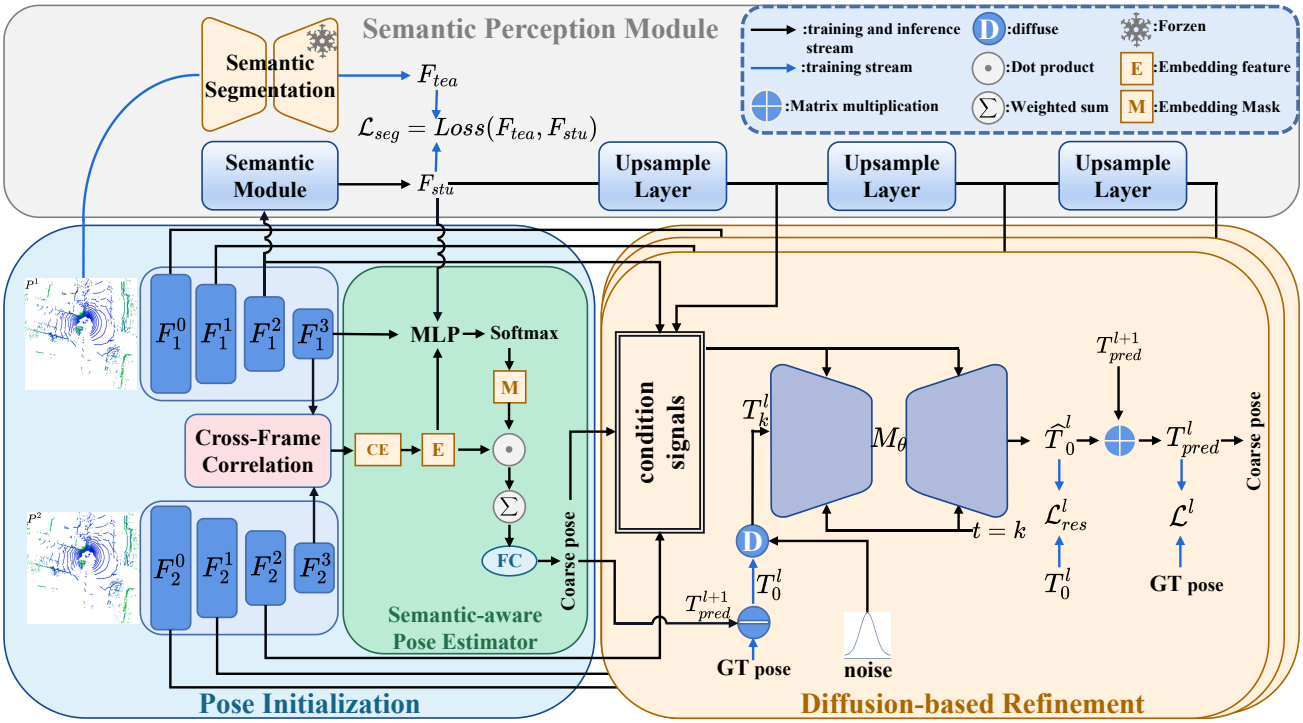


Figure 2. **The overall structure of DiffLO.** The proposed pipeline comprises three core components: **Pose Initialization**, **Semantic Perception Module**, and **Diffusion-based Pose Refinement**. First, hierarchical point features are extracted from the input point cloud. Then, we enable the network to acquire semantic understanding through a knowledge distillation approach. Following this, an initial coarse pose estimate is generated. Finally, the Diffusion-based Pose Refinement module applies iterative refinement layers, conditioned on various signals, to improve the initial pose estimate, ultimately producing a more accurate and robust final pose. To optimize inference efficiency, the semantic segmentation component is discarded after the training phase.

3.3. Semantic-Aware Pose Estimator

We enhance the pose estimation method from the PWCLO-Net [42] by incorporating semantic features into the pose estimation process. Taking the l^{th} layer as example.

$$E^l = MLP(F^l \oplus F^l_{stu} \oplus UE^l \oplus CE^l), \tag{3}$$

$$M^l = \sigma(MLP(F^l \oplus F^l_{stu} \oplus UM^l \oplus E^l)), \tag{4}$$

where $\oplus$ is the concatenation and $\sigma(\cdot)$ denotes the softmax operation. F^l is the point cloud feature from l layer, CE^l is generated by cross-frame correlation [6, 43]. F^l_{stu} , UE^l and UM^l are the results of upsampling the semantic feature F^{l+1}_{stu}, the embedding feature E^{l+1} and the embedding mask M^{l+1} respectively. The embedding features $E = \{e_i | e_i \in \mathbb{R}^C\}^n_{i=1}$, the embedding mask $M = \{m_i | m_i \in \mathbb{R}^C\}^n_{i=1}$. The pose is then generated by:

$$q = \frac{FC(\sum_{i=1}^n e_i \odot m_i)}{|FC(\sum_{i=1}^n e_i \odot m_i)|} \tag{5}$$

$$t = FC(\sum_{i=1}^n e_i \odot m_i) \tag{6}$$

where $FC(\cdot)$ denotes the fully connected layer and $\odot$ represents dot product. Following [42], the embedding features E, embedding mask M and semantic features are also propagated to denser layers for optimization.

The coarse-level point cloud is sparse and has low resolution, which can lead to incorrect matches and inaccurate initial pose estimates. Inspired by recent success and denoising properties of the diffusion model [11], we design a novel diffusion-based pose refinement module to refine the coarse pose progressively.

3.4. Diffusion-based Pose Refinement

In this section, we formulate the coarse pose residual as the diffusion latent variable, and residual pose are generated from the reverse process of diffusion model iteratively.

During the training stage, the diffusion model is trained to learn the underlying distribution of residual poses by recovering the ground truth residual pose from its corrupted version. The ground truth residual pose is calculated as follow:

$$T_0 = T_{gt} \cdot T^{-1}_{pred}, \tag{7}$$

where T_0 is the ground truth residual pose, T_{gt} is the ground truth pose and T_{pred} is the coarse pose generated by the

previous layer or the initial pose.

Specifically, in each training iteration, a random diffusion step, denoted as $k \in \{1, 2, \ldots, K\}$, is chosen. With a predefined variance schedule $\beta_1, \ldots, \beta_k$, we introduce noises to T_0, resulting in the noisy residual pose T_k.

$$q(T_k|T_0) = \mathcal{N}(T_k; \sqrt{\overline{\alpha}_k}T_0, (1 - \overline{\alpha}_k)\mathrm{I}) \tag{8}$$

$$T_k = \sqrt{\overline{\alpha}_k}T_0 + \sqrt{1 - \overline{\alpha}_k}\epsilon, \epsilon \sim \mathcal{N}(0, \mathrm{I}) \tag{9}$$

denoting $\alpha_k = 1 - \beta_k$ and $\overline{\alpha}_k = \Pi_{i=1}^{k}\alpha_i$. Eventually,the ground truth residual pose T_0 is turned into Gaussion noise when k is large enough.

During the inference stage, the denoise model approximates the distribution $p(T_0|C)$ as the reversed diffusion process of gradually cleaning T_k . Instead of predicting ϵ as formulated by [12],we follow [34, 38] and predict the signal itself. Also,we constrain the generation diversity and control the reversed diffusion process by the condition information C. In this way, we utilize a denoising network to predict the residual pose $\hat{T}_0$ from noisy input T_k, time step $t = k$, and condition C.

$$\hat{T}_0 = \mathcal{M}_\theta(T_k, t, C), \tag{10}$$

where $\mathcal{M}_\theta(\cdot, \cdot, \cdot)$ is the denoising network. The training objective is to minimize the following loss:

$$\mathcal{L}_{res} = \mathbb{E}_{T_0,t}, [\|t_{gt} - t_{pred}\|_2 + \|q_{gt} - q_{pred}\|_2], \tag{11}$$

where t_{gt} and q_{gt} are the translation vector and quaternion from T_0, and t_{pred} and q_{pred} are their corresponding predicted values from $\hat{T}_0$.

Design of Condition Signal. Inspired by [42, 43], we introduce the geometric feature GE extracted from PC_1, along with a cost volume embedding CE constructed using cross-frame correlation [6], with the corresponding hierarchical point cloud features as input, and residual pose embeddings $PE = T_k \cdot PC_1 - PC_1$ and the semantic feature F_{stu} as conditional signals to constrain generation diversity. The condition signal is as follow:

$$C = GE \oplus CE \oplus PE, \tag{12}$$

where $\oplus$ is the concatenation operation.

We employ the augmented version [6] with Gate Recurrent Unit (GRU) to process the condition signal C. This approach allows for the extraction of essential features from C to generate the augmented condition signal C'. Subsequently, the semantic-aware pose estimator in Sec. 3.3 is introduced to generate the residual pose $\hat{T}_0$. Here, C' are treated as the cost volume embedding CE in Eq. (3). Taking the l^{th} refinement layer as example, the refined pose T^l_{pred} is caculated as follow :

$$T^l_{pred} = \hat{T_0}^{\,l} \cdot T^{l+1}_{pred}, \tag{13}$$

the refined pose T^l_{pred} will be the input to $l-1^{th}$ refinement layer.

3.5. Training Loss

The network supervises the pose outputs from each level, along with the residual poses predicted by the diffusion model and the semantic features derived from the Semantic Module. Drawing from previous deep odometry research [21, 42], the training loss function for the odometry at the l-th level is as follows:

$$\begin{aligned}\mathcal{L}^l = &\left\|t_{gt} - t^l\right\| exp(-w_t) + w_t \\ &+ \left\|q_{gt} - \frac{q^l}{\|q^l\|}\right\|_2 exp(-w_q) + w_q,\end{aligned} \tag{14}$$

where $\|\cdot\|$ and $\|\cdot\|_2$ denotes the $\ell 1$-norm and the $\ell 2$-norm respectively. t_{gt} and q_{gt} are the ground-truth translation vector and quaternion respectively. t^l and q^l are predicted pose from each level. w_t and w_q are two learnable parameters, which are introduced to eliminate the differences in scale and units between the translation vector and the quaternion. Then, a multi-scale supervised approach is adopted. The total training loss function $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}_{kd} + \sum_{l=0}^{K} \lambda^l \times (\mathcal{L}^l + \mathcal{L}^l_{res}), \tag{15}$$

where K is the total number of warp-refinement levels and λ^l denotes the weight of the l-th level.

4. Experiment

4.1. KITTI Odometry Dataset

The KITTI odometry dataset [7] is a widely used standard dataset for evaluating visual odometry and SLAM algorithms. It includes multiple driving scenarios, encompassing urban, rural, and highway environments, with variations in weather and lighting conditions. The dataset consists of 22 sequences of LiDAR point clouds and their corresponding stereo images. In our experiments, we utilize the Velodyne LiDAR point clouds provided in the dataset. Since the ground truth poses (trajectories) are only available for sequences 00 to 10, we use these sequences for both training and testing.

4.2. Implement details

In the training and evaluation process, N points are randomly sampled from the point clouds of two frames separately. It is not necessary for the original input point clouds to have the same number of points. In the proposed network, N is set to 8192. All training and evaluation experiments are conducted on a single NVIDIA RTX 3090 GPU. The Adam optimizer is employed with parameters $\beta 1$ = 0.9 and $\beta 2$ = 0.999.The initial learning rate is set to 0.001 and is reduced by a factor of 0.8 every 8 epochs until it reaches 0.000001.The initial values of the trainable parameters w_t

Method	00		01		02		03		04		05		06		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}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
ICP-po2po	6.88	2.99	11.21	2.58	8.21	3.39	11.07	5.05	6.64	4.02	3.97	1.93	1.95	1.59	5.17	3.35	10.04	4.93	6.93	2.89	8.91	4.74	7.763	3.978
ICP-po2pl	3.80	1.73	13.53	2.58	9.00	2.74	2.72	1.63	2.96	2.58	2.29	1.08	1.77	1.00	1.55	1.42	4.42	2.14	3.95	1.71	6.13	2.60	4.013	1.968
GICP [36]	1.29	0.64	4.39	0.91	2.53	0.77	1.68	1.08	3.76	1.07	1.02	0.54	0.92	0.46	0.64	0.45	1.58	0.75	1.97	0.77	1.31	0.62	1.375	0.648
CLS [41]	2.11	0.95	4.22	1.05	2.29	0.86	1.63	1.09	1.59	0.71	1.98	0.92	0.92	0.46	1.04	0.73	2.14	1.05	1.95	0.92	3.46	1.28	2.148	0.995
Full LOAM [50]	0.78	0.53	1.43	0.55	0.92	0.55	0.86	0.65	0.71	0.50	0.57	0.38	0.65	0.39	0.63	0.50	1.12	0.44	0.77	0.48	0.79	0.57	0.828	0.498
Full A-LOAM	0.76	0.31	1.97	0.50	4.53	1.45	0.93	0.49	0.62	0.39	0.48	0.25	0.61	0.28	0.43	**0.26**	**1.06**	**0.32**	0.73	0.31	1.02	0.40	0.810	0.323
LO-Net [21]	1.47	0.72	1.36	0.47	1.52	0.71	1.03	0.66	0.51	0.65	1.04	0.69	0.71	0.50	1.70	0.89	2.12	0.77	1.37	0.58	1.80	0.93	1.748	0.793
PWCLO-Net [42]	0.78	0.42	0.67	0.23	0.86	0.41	0.76	0.44	0.37	0.40	0.45	0.27	**0.27**	0.22	0.60	0.44	1.26	0.55	0.79	0.35	1.69	0.62	1.085	0.490
DELO [1]	1.43	0.81	2.19	0.57	1.48	0.52	1.38	1.10	2.45	1.70	1.27	0.64	0.83	0.35	0.58	0.41	1.36	0.64	1.23	0.57	1.53	0.90	1.175	0.630
TransLO [23]	0.85	0.38	1.16	0.45	0.88	0.34	1.00	0.71	0.34	0.18	0.63	0.41	0.73	0.31	0.55	0.43	1.29	0.50	0.95	0.46	1.18	0.61	0.993	0.500
EfficientLO [44]	0.83	0.33	0.55	0.21	**0.71**	**0.25**	**0.49**	**0.38**	**0.22**	**0.11**	0.34	**0.21**	0.36	0.24	0.46	0.38	1.14	0.41	0.78	0.33	0.80	0.46	0.795	0.395
Ours	**0.60**	**0.27**	**0.36**	**0.15**	**0.71**	0.26	0.57	0.40	0.30	0.17	**0.33**	0.22	0.28	**0.17**	**0.37**	0.27	1.12	0.44	**0.68**	**0.28**	**0.66**	**0.32**	**0.708**	**0.328**

Table 1. The LiDAR odometry experiment results on KITTI odometry dataset [7]. 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. † means the testing sequences. LOAM and A-LOAM is complete SLAM system,including back-end optimization. The best result for each sequence is bold, and the second best is underlined.

Method	07†		08†		Mean	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
LodoNet [53]	1.86	1.64	2.04	0.97	1.950	1.305
Ours	**0.37**	**0.27**	**1.12**	**0.44**	**0.745**	**0.355**

Table 2. Comparison with learning-based odometry LodoNet on KITTI 07-08 sequences.

Method	09†		10†		Mean	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Nubert et al. [30]	1.54	0.68	1.78	0.69	1.66	0.685
H-VLO [2]	1.89	0.34	1.39	0.52	1.64	0.430
Ours	**0.68**	**0.28**	**0.66**	**0.32**	**0.67**	**0.300**

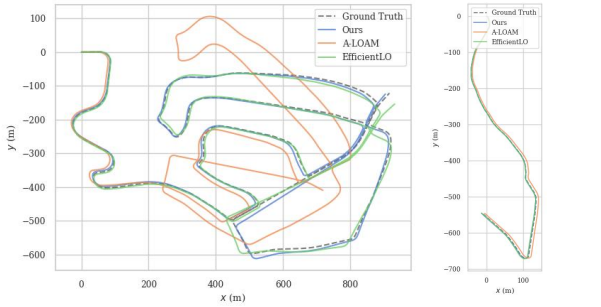
Table 3. Comparison with learning-based odometry on KITTI 09-10 sequences.

and w_q are set to 0.0 and -2.5, respectively, as indicated in formula Eq. (14). For formula Eq. (15), the parameters are set as follows: $\lambda_3 = 1.6$, $\lambda_2 = 0.8$, $\lambda_1 = 0.4$, $\lambda_0 = 0.2$, and $L = 4$. The batch size is 12.

4.3. Comparison with The State-of-the-Art

We compare our network with the state-of-the-art on KITTI dataset, including both classic methods and learning-based ones. Since existing training and testing sequence settings are inconsistent in different methods, we test and evaluate our framework accordingly for a fair comparison.

00-06 as training sequences and 07-10 as testing sequences. Quantitative results are listed in Tab. 1. The traditional methods we compare include ICP-point2point (ICP-po2po), ICP-point2plane (ICP-po2pl), GICP [36], CLS [41], which rely on iterative point cloud alignment. LOAM [50], its enhanced version A-LOAM is also included. While LOAM and A-LOAM have been widely recognized for their robust performance on the KITTI odometry benchmark, our method achieves superior accuracy on most sequences. As shown in Tab. 1, our approach outperforms Full A-LOAM,



(a) 2D Trajectory Plots of Seq.02 (b) 2D Trajectory Plots of Seq.10

Figure 3. 2D trajectories of A-LOAM, EfficientLO, and our proposed method on KITTI sequences 02 and 10 with ground truth. It can be observed that our method performs the best.

which has the mapping optimization. Additionally, the learning-based LiDAR odometry methods we compare include LO-Net [21], DELO [1], PWCLO-Net [42], TransLO [23], EfficientLO [44]. Compared to the current state-of-the-art method EfficientLO, our method DiffLO achieves an average improvement of 10.9% in transition accuracy and 16.9% in rotation accuracy.

00-06 and 09-10 as training sequences and 07-08 as testing sequences. We compare with a recent method LodoNet [52] in Tab. 2. Our method is trained on 00-06 sequences while LodoNet are trained on 00-06 and 09-10 sequences,even so, our method performs better than LodoNet on the test sequences.

00-08 as training sequences and 09-10 as testing sequences. We compare with recent method [2, 30]. Our method is trained on 00-06 sequences while others are trained on 00-08 sequences. As illustrated in Tab. 3, both our translation and rotation errors are smaller than theirs. Although our approach utilizes only LiDAR data, we achieve better results than the multi-modal method H-VLO.

4.4. Ablation Study

In order to analyze the effectiveness of each module, we remove or change components of our model to do the abla-

	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}
(a)Semantic	w/o SPM	0.52	0.46	1.19	0.53	**0.53**	**0.18**	0.93	0.42	0.793	0.398
	Ours (full, with SPM)	**0.37**	**0.27**	**1.12**	**0.44**	0.68	0.28	**0.66**	**0.32**	**0.708**	**0.328**
(b)Diffusion	w/o diffusion	4.83	3.82	6.77	3.05	5.29	2.22	7.09	3.97	5.995	3.265
	with trans-based M_θ [40]	0.39	0.34	1.36	0.55	0.73	0.29	1.07	0.48	0.888	0.415
	with ptrans-based M_θ [51]	0.45	0.28	1.19	0.45	**0.54**	**0.22**	0.72	0.41	0.725	0.340
	Ours (GRU-based M_θ [6])	**0.37**	**0.27**	**1.12**	**0.44**	0.68	0.28	**0.66**	**0.32**	**0.708**	**0.328**
(c)Condition	w/o cost volume	0.45	0.35	1.20	0.49	0.70	0.28	1.09	0.95	0.860	0.518
	w/o geometry feature	0.40	**0.27**	1.18	0.47	**0.61**	0.24	0.96	0.46	0.787	0.357
	w/o semantic feature	0.47	0.31	1.33	0.56	0.75	**0.23**	0.64	0.48	0.798	0.395
	w/o coarse pose embedding	0.45	0.34	1.28	0.56	0.64	0.25	**0.63**	0.33	0.750	0.370
	Ours (full, with all conditions)	**0.37**	**0.27**	**1.12**	**0.44**	0.68	0.28	0.66	**0.32**	**0.708**	**0.328**

Table 4. The ablation study results of LiDAR odometry on KITTI odometry dataset[7]. The best performance of each sequence is bold.

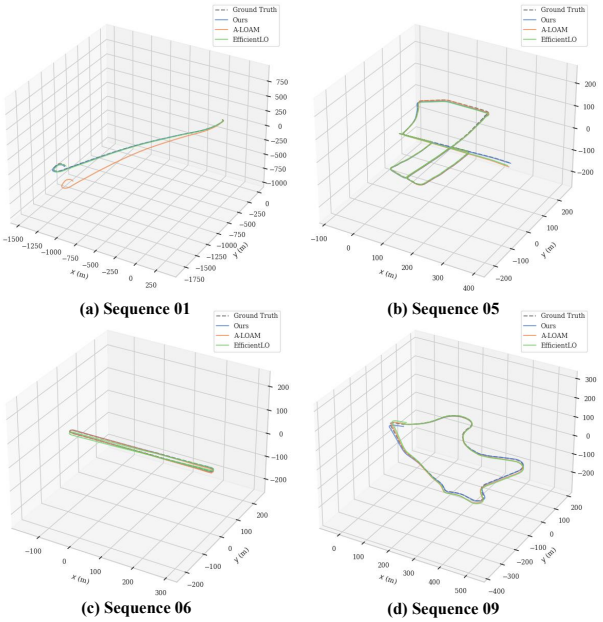


Figure 4. 3D trajectories of A-LOAM , EfficientLO and our proposed method on KITTI sequences 01, 05, 06 and 09 with ground truth. Ours trajectory is more accurate than EfficientLO and the Full A-LOAM, which has the mapping optimization.

tion studies on KITTI odometry dataset (00-06 as training sequences and 07-10 as testing sequences).

Study on Semantic Perception Module. We conducted ablation experiments to demonstrate the significance of the Semantic Perception Module (SPM) mentioned in Sec. 3.2. As shown in Tab. 4, SPM achieves an average improvement of 10.7% in transition accuracy and 17.6% in rotation accuracy. This indicates that SPM enhances accuracy by embedding semantic understand to the network, which helps the network to focus more on the objects in the scene that play a key role in pose estimation.

Study on Diffusion Refinement. We conducted an ablation study by comparing results with and without the diffusion-based refinement process (w/o diffusion). As shown in Tab. 4, the diffusion-based pose refinement significantly improved performance, reducing the average translation error by 88.2% (from 5.995% to 0.708%) and the average rotation error by 90.0% (from 3.265°/100m to 0.328°/100m). Additionally, we explored the performance of different denoising networks, including a transformer-based [40] network (trans-based), a point transformer-based [51] network (ptrans-based), and GRU-based [6] network. The results demonstrate the advantages of our denoising network over alternative approaches.

Study on Condition signals. To evaluate the effectiveness of the diffusion condition signals, we conducted an ablation study by individually removing each designed condition signal, as shown in Tab. 4. Removing the cost volume led to a performance degradation (with translation error increasing by 21.5% and rotation error by 57.9%), since cost volume captures the precise per-point correlation between two frames. The semantic feature also plays a crucial role in guiding diffusion generation, as it provides scene understanding. Consequently, its removal resulted in a significant drop in performance (led to a 12.7% increase in translation error and a 20.4% increase in rotation error). Additionally, removing the coarse pose embedding led to a decline in performance, as coarse pose embedding provides essential state information, which is closely related to the residual pose generated by the model. Finally, the absence of geometry features as guidance also resulted in a noticeable performance drop.

4.5. Visualization

We visualize the trajectory of our network and analyze the benefits of Semantic Perception Module in this section.

Visualization of Trajectory. The qualitative results are shown in Fig.3, 4 and 5. We compared our method with full A-LOAM, which includes mapping optimization, and EfficientLO [44]. Notably, in Seq.02 of Fig. 3, A-LOAM exhibits significant drift due to the presence of repetitive

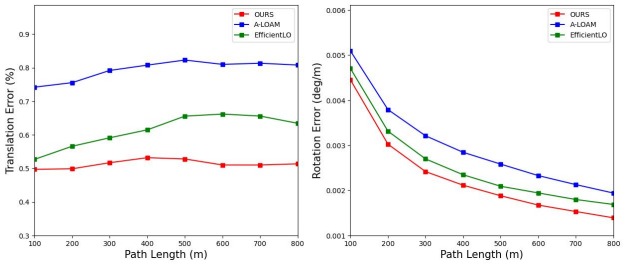


Figure 5. Average translational and rotational error on KITTI sequences 00-10 on all possible subsequences in the length of 100, 200, ..., 800m. Our method has the best performance.

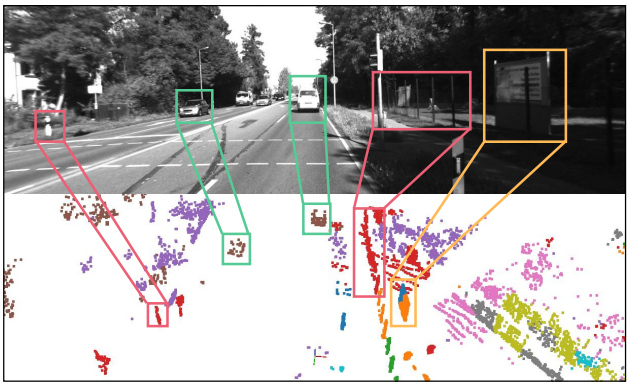


Figure 6. The visualization of semantic features is derived from the complete LiDAR point cloud of the initial frame. Additionally, the grayscale image captured from the camera view corresponds to the semantic segmentation image displayed below. Bounding boxes are utilized to delineate specific object categories.

structures and patterns in the scene, which impedes the algorithm's ability to accurately estimate pose. These figures show that our odometry can track the trajectory of the ground truth fairly well. At the same time, our method demonstrates the best average evaluation performance compared to A-LOAM and EfficientLO.

Visualization of Semantic feature. The proposed network employs an embedding feature of 8,192 points to compute the pose transformation in the final pose output layer. Consequently, we visualize the semantic features of these 8,192 points. As illustrated in Fig. 6, vehicles, street lamps, fences, billboards, and other objects in the scene are segmented. This demonstrates that the Semantic Perception Module empowers the network with the capability of semantic understanding.

Visualization of embedding mask. We visualize the embedding mask of the 8,192 points in the final pose output layer. As shown in Fig. 7, the mask of DiffLO integrated with the Semantic Perception Module (SPM) allocates weights more accurately compared to the mask of DiffLO without SPM. Points sampled from static and structured rigid objects, such as buildings, fences, and pole-like object, exhibit higher weights, while points from dynamic

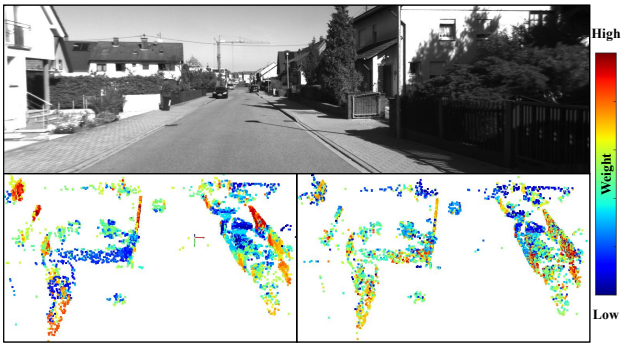


Figure 7. The visualization of the embedding mask is based on the complete LiDAR point cloud of the initial frame. In the lower-left corner, the heatmap represents the mask of DiffLO integrated with the SPM. Conversely, the lower-right corner displays the heatmap for the mask of DiffLO without SPM. Above these visualizations, the corresponding grayscale image of the point cloud is presented.

and low-texture objects, including cars, shrubs, and trees, receive lower weights. This demonstrates that the Semantic Perception Module (SPM) effectively helps the network focus on objects in the scene that are crucial for accurate pose estimation.

4.6. Runtime Analysis

Efficiency is another extremely significant factor in real-time SLAM systems. The LiDAR points in the KITTI dataset are captured at a 10Hz frequency. Our method has only 76.9 ms inference time, which satisfy the real-time application requirements (under 100 ms).

5. Conclusion

In this paper, we introduce a diffusion model-based odometry network augmented with semantic awareness, targeting large-scale LiDAR odometry. Our approach innovatively applies knowledge distillation to transfer semantic information into the odometry network, thereby enhancing the network's semantic understanding. This allows the network to effectively mitigate the challenges posed by dynamic objects, repetitive patterns, and low-texture environments. Additionally, we incorporate a diffusion model conditioned on pose-related features to refine pose estimation. To the best of our knowledge, this is the first instance of utilizing diffusion models in odometry tasks. We evaluated our framework on the KITTI odometry dataset, and our method achieved state-of-the-art performance while ensuring the real-time requirement of pose prediction.

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

References

[1] Sk Aziz Ali, Djamila Aouada, Gerd Reis, and Didier Stricker. Delo: Deep evidential lidar odometry using partial optimal transport. In *ICCV*, pages 4517–4526, 2023. 2, 6

[2] Eren Aydemir, Naida Fetic, and Mustafa Unel. H-vlo: hybrid lidar-camera fusion for self-supervised odometry. In *IROS*, pages 3302–3307. IEEE, 2022. 6

[3] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *RSS*, page 59, 2018. 2

[4] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 1

[5] Guibin Chen, Bosheng Wang, Xiaoliang Wang, Huanjun Deng, Bing Wang, and Shuo Zhang. Psf-lo: Parameterized semantic features based lidar odometry. In *ICRA*, pages 5056–5062. IEEE, 2021. 2

[6] Wencan Cheng and Jong Hwan Ko. Multi-scale bidirectional recurrent network with hybrid correlation for point cloud based scene flow estimation. In *ICCV*, pages 10041–10050, 2023. 3, 4, 5, 7

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32 (11):1231–1237, 2013. 2, 5, 6, 7

[8] Linrui Gong, Jiuming Liu, Junyi Ma, Lihao Liu, Yaonan Wang, and Hesheng Wang. Eadreg: Probabilistic correspondence generation with efficient autoregressive diffusion model for outdoor point cloud registration. *arXiv preprint arXiv:2411.15271*, 2024. 2

[9] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 2

[10] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 4

[12] Jonathan Ho, AjayN. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 5

[13] Hanjiang Hu, Zhijian Qiao, Ming Cheng, Zhe Liu, and Hesheng Wang. Dasgil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE Transactions on Image Processing*, 30:1342–1353, 2020. 1

[14] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *PAMI*, 44(11):8338–8354, 2021. 2

[15] Haobo Jiang, Mathieu Salzmann, Zheng Dang, Jin Xie, and Jian Yang. Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation. *NeurIPS*, 36, 2024. 2, 3

[16] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, pages 17545–17555, 2023. 3

[17] Felix Järemo Lawin, Martin Danelljan, Patrik Tosteberg, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Deep projective 3d semantic segmentation. In *Computer Analysis of Images and Patterns: 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*, pages 95–107. Springer, 2017. 2

[18] Daegyu Lee, Hyunwoo Nam, and D Hyunchul Shim. Eliot: End-to-end lidar odometry using transformer framework. *arXiv preprint arXiv:2307.11998*, 2023. 2

[19] Jinkyu Lee, Muhyun Back, Sung Soo Hwang, and Il Yong Chun. Improved real-time monocular slam using semantic segmentation on selective frames. *TITS*, 24(3):2800–2813, 2023. 2

[20] Lin Li, Xin Kong, Xiangrui Zhao, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Sa-loam: Semantic-aided lidar slam with loop closure. In *ICRA*, pages 7627–7634. IEEE, 2021. 2

[21] 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. 2, 5, 6

[22] Wen Li, Yuyang Yang, Shangshu Yu, Guosheng Hu, Chenglu Wen, Ming Cheng, and Cheng Wang. Diffloc: Diffusion model for outdoor lidar localization. In *CVPR*, pages 15045–15054, 2024. 2, 3

[23] Jiuming Liu, Guangming Wang, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Translo: A window-based masked point transformer framework for large-scale lidar odometry. In *AAAI*, pages 1683–1691, 2023. 2, 6

[24] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: an efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8451–8460, 2023. 2

[25] Jiuming Liu, Guangming Wang, Weicai Ye, Chaokang Jiang, Jinru Han, Zhe Liu, Guofeng Zhang, Dalong Du, and Hesheng Wang. Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In *CVPR*, pages 15109–15119, 2024. 2, 3

[26] Jiuming Liu, Dong Zhuo, Zhiheng Feng, Siting Zhu, Chensheng Peng, Zhe Liu, and Hesheng Wang. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment. In *European Conference on Computer Vision*, pages 475–493. Springer, 2025. 2

[27] Zhe Liu, Shunbo Zhou, Chuanzhe Suo, Peng Yin, Wen Chen, Hesheng Wang, Haoang Li, and Yun-Hui Liu. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *ICCV*, pages 2831–2840, 2019. 1

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

[29] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220. IEEE, 2019. 2

[30] Julian Nubert, Shehryar Khattak, and Marco Hutter. Self-supervised learning of lidar odometry for robotic applications. In *ICRA*, pages 9601–9607. IEEE, 2021. 6

[31] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing icp variants on real-world data sets: Open-source library and experimental protocol. *Autonomous robots*, 34:133–148, 2013. 1

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2

[33] 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

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 5

[35] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, pages 3212–3217. IEEE, 2009. 1

[36] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *RSS*, page 435. Seattle, WA, 2009. 6

[37] 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. 2

[38] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model, 2022. 5

[39] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6411–6420, 2019. 2

[40] A Vaswani. Attention is all you need. *NeurIPS*, 2017. 7

[41] Martin Velas, Michal Spanel, and Adam Herout. Collar line segments for fast odometry estimation from velodyne point clouds. In *ICRA*, pages 4486–4495. IEEE, 2016. 6

[42] 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, 2, 4, 5, 6

[43] Guangming Wang, Yunzhe Hu, Zhe Liu, Yiyang Zhou, Masayoshi Tomizuka, Wei Zhan, and Hesheng Wang. What matters for 3d scene flow network. In *ECCV*, pages 38–55. Springer, 2022. 4, 5

[44] Guangming Wang, Xinrui Wu, Shuyang Jiang, Zhe Liu, and Hesheng Wang. Efficient 3d deep lidar odometry. *PAMI*, 45 (5):5749–5765, 2022. 2, 6, 7

[45] Hesheng Wang, Dongliang Zheng, Jingchuan Wang, Weidong Chen, and Jianjun Yuan. Ego-motion estimation of a quadrotor based on nonlinear observer. *IEEE/ASME Transactions on Mechatronics*, 23(3):1138–1147, 2018. 1

[46] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, pages 1887–1893. IEEE, 2018. 2

[47] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2019. 3

[48] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *ECCV*, pages 88–107. Springer, 2020. 3

[49] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *ECCV*, pages 677–695. Springer, 2022. 2

[50] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous robots*, 41:401–416, 2017. 6

[51] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 7

[52] Ce Zheng, Yecheng Lyu, Ming Li, and Ziming Zhang. Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation. In *ACMMM*, pages 2391–2399, 2020. 2, 6

[53] Ce Zheng, Yecheng Lyu, Ming Li, and Ziming Zhang. Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation. In *ACMMM*, pages 2391–2399, 2020. 6

[54] Xin Zheng and Jianke Zhu. Efficient lidar odometry for autonomous driving. *IEEE Robotics and Automation Letters*, page 8458–8465, 2021. 1

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

[56] Pengwei Zhou, Xuexun Guo, Xiaofei Pei, and Ci Chen. T-loam: Truncated least squares lidar-only odometry and mapping in real time. *TGRS*, 60:1–13, 2022. 2

[57] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, pages 9939–9948, 2021. 2

[58] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, pages 12786–12796, 2022. 1