DELO: Deep Evidential LiDAR Odometry using Partial Optimal Transport

Sk Aziz Ali[⋆†] Djamila Aouada[†] Gerd Reis[⋆] Didier Stricker[⋆]

[†]SnT, University of Luxembourg [⋆]German Research Center for Artificial Intelligence (DFKI)

Abstract

Accurate, robust, and real-time LiDAR-based odometry (LO) is imperative for many applications like robot navigation, globally consistent 3D scene map reconstruction, or safe motion-planning. Though LiDAR sensor is known for its precise range measurement, the non-uniform and uncertain point sampling density induce structural inconsistencies. Hence, existing supervised and unsupervised point set registration methods fail to establish one-to-one matching correspondences between LiDAR frames. We introduce a novel deep learning-based real-time (∼35-40ms per frame) LO method that jointly learns accurate frame-to-frame correspondences and model's predictive uncertainty (PU) as evidence to safe-guard LO predictions. In this work, we propose (i) partial optimal transportation of LiDAR feature descriptor for robust LO estimation, (ii) joint learning of predictive uncertainty while learning odometry over driving sequences, and (iii) demonstrate how PU can serve as evidence for necessary pose-graph optimization when LO network is either under or over confident. We evaluate our method on KITTI dataset and show competitive performance, even superior generalization ability over recent state-of-the-art approaches. Source codes are available.

1. Introduction

In partial scan-to-scan alignment setting, LiDAR odometry (LO) is defined as the problem of estimating the 6-DoF ego-motion $\mathbf{T}_f \in \mathrm{SE}(3)$, *i.e.*, the pose of the LiDAR sensor at frame f relative to the pose at previous frame, given two consecutive *undistorted* scans. This step serves as the backbone of most methods for robotic motion/path planning [32], navigation [27], simultaneous localization and mapping (SLAM) [9], and many other complex scene reconstruction [45] tasks.

Unlike inertial or wheel odometry [48] using IMU sensors, LiDAR-only odometry estimation is more challenging. This is due to four primary reasons – ***(1)*** non-uniform point sampling *density* of the sensor induces structural imbalance into the scan, ***(2)*** change of speed in the moving sensors results into an out of order distribution (OOD) of relative sensor motion, ***(3)*** scanned points that are acquired by LiDAR sensor in consecutive frames include large number of false positives and *uncertain* matching correspondences, and finally ***(4)*** previous three factors *i.e. density, distribution, and uncertainty* alleviate the solution multiplicity [16] problem significantly. For these reasons, the challenges in estimating relative motion of LiDAR sensor are greater than general rigid point set registration (RPSR) methods. Primarily, inconsistent drift or velocity model of ego-vehicle, dynamic objects in the scene, and cumulative error propagation due to susceptible pose predictions of all the intermediate frames are the extra difficulties for LO methods. For these reasons, robust LiDAR point correspondence matching and self-supervised LO estimation is still an open problem.

In this paper, we propose a Deep Evidential LiDAR Odometry (DELO) to overcome the aforementioned challenges using a unified multi-task learning approach (see Figure 1). In summary, our main **contributions** are – ***(i)*** designing a neural network for frame-to-frame LiDAR descriptor matching (LDM) (Sec. 3.1) using partial optimal transport (POT) [17, 40, 12] plan, termed as POT-LDM. This network assigns a higher weights to inliers, even if they are small in numbers between two LiDAR frames. This is a natural way to tackle of the solution multiplicity [16] problem in LiDAR feature matching. Next, ***(ii)*** a network that learns predictive uncertainty for evidential pose estimation, termed as PU-EPE (Sec. 3.2). We describe how the learned uncertainty over predicted poses are approximately equivariant along different transformation axes. This is an elegant way to classify under-confident, confident, and over-confident LO predictions. Finally, ***(iii)*** we show how the pose uncertainty can act as evidences of anomaly related to LO estimation. Herein, the dynamic pose refinement (Sec. 3.3) prevents further propagation of prediction errors.

2. Related Work

Deep Learning-based Point Set Registration. A set of purely geometric rigid point set registration (RPSR) algorithms [3, 2, 42, 11, 22, 48, 29, 37] can be used for

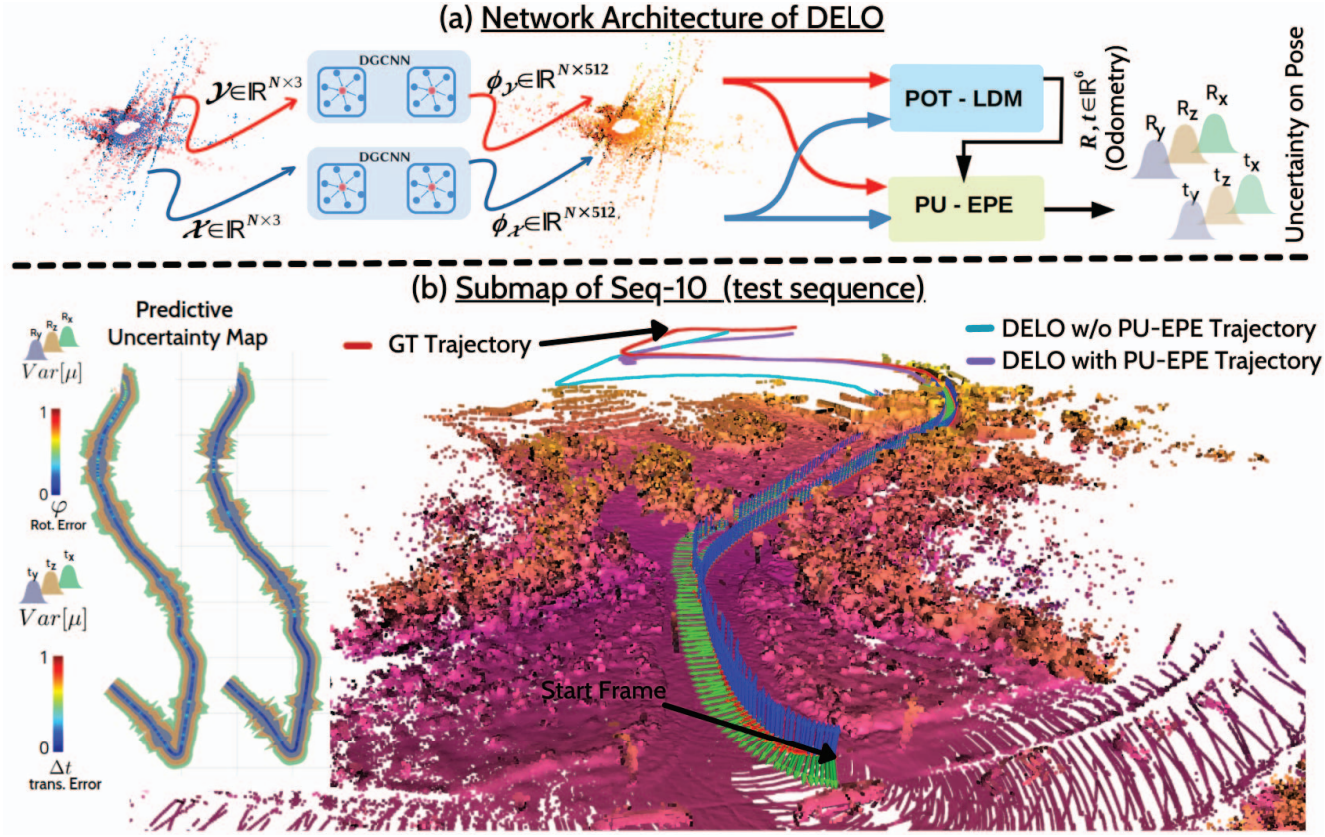


Figure 1. ***(a)* Overview of DELO Network:** Given sequential point clouds of input frames at different positions, DELO applies DGCNN [42] as backbone encoder to obtain a point-wise feature embedding. Then it simultaneously aligns the frames using the Partial Optimal Transport plan for LiDAR Descriptor Matching (POT-LDM), and estimates the Predictive Uncertainty for the Evidential Pose Estimation (PU-EPE). With the help of PU estimates pose-graphs are refined. ***(b)*** This part depicts a sub-map between frame 1 to 300 of KITTI test sequence-10 with all outputs from DELO.

LO estimation. Among them, only a few methods [2, 11, 28, 29, 42, 43] propose deep-learning based approaches, and even fewer methods [2, 42] can infer in real-time. Above all, inhomogeneous distribution of LiDAR points raises a common problem for all LO and registration methods to find one-to-one matching correspondences. For instance, ICP [7], ICP$_\perp$ [35], CPD [30], and other classical approaches [39, 49] for RPSR, except FGA [3], all perform poorly on LiDAR scans. More recently, the deep neural network (DNN) for point set registration – DCP [42], RPSRNet [2], DeepVCP [29] and DGR [11], appear as benchmarks for frame-to-frame registration of LiDAR scans. These methods [42, 2, 29, 11] are faster and perform better than classical techniques [39, 3, 49, 7]. Where RPSRNet (with ∼20ms inference speed) proposes a novel hierarchical representation for inhomogeneous point cloud data, DGR (with ∼700ms inference speed) combines a compact geometric feature map with weighted Orthogonal Procrustes (OP) [19] for effective correspondence matching. DCP [42], a neural version of ICP [7], is the first method to use transformer network [5]. It computes a 'doubly stochastic cross-attention score matrix' to find correspondences between two scans. Despite impressive formulation and learning strategy of DCP and its successor PRNet [43], both suffers from well-known solution-multiplicity [16], *i.e.*, spurious association between false feature correspondences.

Deep Learning-based LiDAR Odometry. Classical ICP-based methods [47, 38, 14, 20] and few carefully designed deep-learning-based methods [41, 10, 24, 31, 25] span the baselines for LiDAR odometry. In our knowledge, we see many of these methods [24, 25, 10, 14, 20, 47, 38] convert 3D LiDAR scans to 2D range images and therefore undermine the vertical pose-drift by scaling it only to few pixels. LO-Net [24], PWCLONet [41], and RPSRNet [2] are among the learning-based real-time methods that directly operate on 3D point clouds. LO-Net uses point normal vectors for 'local geometric consistency' and additional mapping network module (*i.e.*, scan-to-map registration) for refined odometry estimation. Similar idea also exists in unsupervised learning [10, 31]. Instead of relying on geometrical features, PWCLONet demonstrates how to hierarchically build a feature pyramid of point motion [26] be-

tween two scans. The main reason behind such choice is to filter small relative motions between dynamically moving objects (*i.e.*, scene-flow) and capturing large ego-motion for odometry. This technique effectively avoid solution-multiplicity [16]. LO-Net (with mapping), RPSRNet, and PWCLONet (with ∼80ms, ∼20ms, and ∼125ms inference speed respectively) all have reported low drift compared to LOAM [47](without mapping).

LO Uncertainty as Evidence. For an end-to-end trainable odometry network, it is difficult to define a direct map that captures a small perturbation in its measurement (*i.e.*, $\mathbf{T}_f$) at the current frame f and readjust the changes for the future measurements (*i.e.*, $\mathbf{T}_{f+1}, \ldots$). This problem is occurs due to unexpected out-of-order distribution (OOD). Therefore, *uncertainty quantification (UQ) over the predicted transformations helps in setting boundary conditions for any downstream pose-based decision making tasks* [1, 21] – *e.g.*, general classification [36], motion forecasting [32], and navigation [27]. A recent deep multi-task learning approach LP2 [32], for joint localization, perception and prediction tasks underpins the importance of UQ in their setup. Deep Evidential Regression (DER) [4] is now a preferred choice for many learning-based navigation models [27, 36] than conventional and computationally costly UQ techniques [34, 1]. To this end, joint learning of LO and PU, and thereafter using such relational model for automatic odometry refinement, remain unexplored.

3. DELO Method Overview

The proposed DELO operates on a sequence of 3D LiDAR scans $\mathcal{S} = \{\mathcal{X}_f\}_{f=1}^{S}$, where any input scan $\mathcal{X}_f$ at frame f is randomly sub-sampled to a fixed N number of points. Thereafter, DELO takes pair-wise source $\mathcal{Y} \in \mathbb{R}^{N\times 3}$ and target $\mathcal{X} \in \mathbb{R}^{N\times 3}$ point clouds as input, and embeds them independently with DGCNN [44] encoder $\phi : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times D}$. We have used $N = 1024$, $D = 512$.

3.1. POT-LDM: Sharp Correspondence Matching

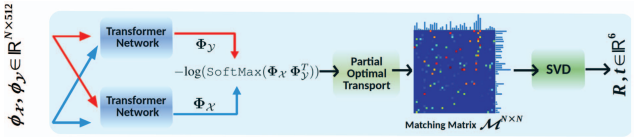


Figure 2. POT-LDM network architecture.

The POT-LDM network, as shown in Figure 2, takes point-wise feature embedding $\phi_{\mathcal{Y}} \in \mathbb{R}^{N\times D}$ and $\phi_{\mathcal{X}} \in \mathbb{R}^{N\times D}$ of source $\mathcal{Y}$ and target $\mathcal{X}$. Next, a transformer network [5] turns the input features into task-specific features [42] using contextual map $\varphi : \mathbb{R}^{N\times D} \times \mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{N\times D}$. This is an asymmetric learnable map for measuring changes between two input embedding tensors $\phi_{\mathcal{Y}}$ and $\phi_{\mathcal{X}}$. Finally, the output of the transformer network,

$$\mathbf{\Phi}_{\mathcal{Y}} = \phi_{\mathcal{Y}} + \varphi(\phi_{\mathcal{Y}}, \phi_{\mathcal{X}}) \text{ and } \mathbf{\Phi}_{\mathcal{X}} = \phi_{\mathcal{X}} + \varphi(\phi_{\mathcal{X}}, \phi_{\mathcal{Y}}), \quad (1)$$

are used for sharp correspondence matching. For this, DCP [42] runs differentiable soft-assignment such that for each point $y_i \in \mathcal{Y}$, a probability vector over $\mathcal{X}$ is assigned as matching measure $m(y_i, \mathcal{X}) = \texttt{SoftMax}(\mathbf{\Phi}_{\mathcal{X}} \mathbf{\Phi}_{y_i}^T)$.

Partial Transportation of Mass as Attention Weights. Due to sensor movement and sparse point clouds in LiDAR-data, we observe only a partial number of points that can be matched between consecutive scans. While point set registration only requires a minimum of three true point correspondences for solving OP [19] problem, it is difficult to know or match true feature correspondences in advance. Therefore, one can assume this as a partial-to-partial sparse rigid point set registration task. We employ partial optimal mass transportation technique [40] for *sharp* point matching instead of its seminal version [12]. In this technique, low transportation cost means input features match closely. We use entropy regularized partial optimal transport [6]

$$\begin{aligned} \boldsymbol{\mathcal{M}} = \arg\min_{\boldsymbol{\mathcal{M}}} \langle \boldsymbol{\mathcal{M}}, \boldsymbol{C} \rangle_F + \lambda \Omega(\boldsymbol{\mathcal{M}}), \\ \text{s.t. } \boldsymbol{\mathcal{M}}\mathbf{1} \leq a, \\ \boldsymbol{\mathcal{M}}^T\mathbf{1} \leq b, \text{ and} \\ \mathbf{1}^T\boldsymbol{\mathcal{M}}^T\mathbf{1} = m \leq \min\{a^T\mathbf{1}, b^T\mathbf{1}\}, \end{aligned} \quad (2)$$

where $\langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{C}} \rangle_F = \text{tr}\left(\boldsymbol{\mathcal{M}}^{\text{T}}\boldsymbol{\mathcal{C}}\right)$ denotes Forbenius norm over matrix dot product, $\mathbf{1} = (1, \ldots, 1)^T$ is a vector of all N elements as 1, $\boldsymbol{\mathcal{M}} \in (\mathbb{R}_+)^{N\times N}$ is the transport matrix, $\boldsymbol{C} \in (\mathbb{R}_+)^{N\times N}$ is the cost matrix, λ is a regularization parameter, $a \in \mathbb{R}_+^{N\times 1}$ and $b \in \mathbb{R}_+^{N\times 1}$ are probability distributions, m is mass to transport and $\Omega(\boldsymbol{\mathcal{M}}) = \sum_{i,j} \boldsymbol{\mathcal{M}}_{i,j} \log(\boldsymbol{\mathcal{M}}_{i,j})$ is the entropic regularization term. The amount of transported mass m between both inputs acts as a control parameter to adjust the 'sharpness' of the correspondence matching. The regularization parameter λ is learned during network training. Each point initially has an equal probability of $a, b = \mathbf{1}/N$. We set the cost matrix $\boldsymbol{C}$ as the negative log-likelihood of the matching probabilities [42] for every point $y_i \in \mathcal{Y}$ with all points in $\mathcal{X}$ such that its matching cost

$$\boldsymbol{C}_{y_i} = -\log(\texttt{SoftMax}(\mathbf{\Phi}_{\mathcal{X}}\, \mathbf{\Phi}_{y_i}^T)). \quad (3)$$

The negative log-likelihood penalizes the outlier points. Algorithm 1 describes entropy regularized POT solution steps following [6] and [17]. It is possible to set different mass initialization and iteration limits to the input of Algorithm 1. An optimal transportation cost is achieved faster and efficiently if one initially sets low masses m and a fewer number of iterations for Algorithm 1. The scores from the matching matrix $\boldsymbol{\mathcal{M}}$ are used to compute the rigid transformation $\mathbf{T} = [\mathbf{R}, \mathbf{t}]$ using weighted-Procrustes [46, 11, 8] with differentiable SVD.

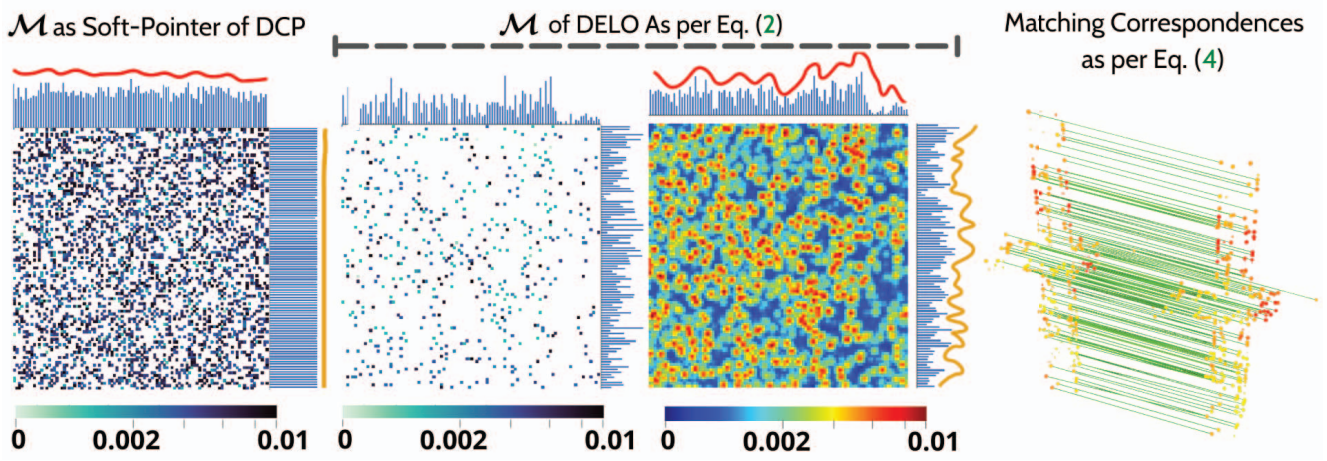


Figure 3. **Matching Matrix Comparison:** DCP [42] Soft-Pointer based matching matrix $\mathcal{M}$ (on the left). The histogram plot over the rows and columns denote sum of probability vectors along the dimensions. Two plots from the right (with different colormaps for visual clarity) denote $\mathcal{M}$ when optimized by POT-LDM.

Algorithm 1: Partial Optimal Transport

Data: Cost matrix $\boldsymbol{C}$, mass m, Iter. ξ, regularizer λ
Result: Partial Optimal Transport Mass $\boldsymbol{\mathcal{M}}$

begin
  $a, b \leftarrow \mathbf{1}/N$ and $\mathbf{1}/N$;
  $\boldsymbol{K} \leftarrow e^{-\boldsymbol{C}/\lambda}$;
  for $i \leftarrow$ **to** ξ **do**
    $\tilde{\boldsymbol{K}} \leftarrow \text{diag}(\min(\frac{a}{\tilde{\boldsymbol{K}}\mathbf{1}}, \mathbf{1}))\boldsymbol{K}$;
    $\hat{\boldsymbol{K}} \leftarrow \tilde{\boldsymbol{K}}\text{diag}(\min(\frac{b}{\tilde{\boldsymbol{K}}^T\mathbf{1}}, \mathbf{1}))$;
    $\boldsymbol{K} \leftarrow \hat{\boldsymbol{K}}\frac{m}{\mathbf{1}^T\hat{\boldsymbol{K}}\mathbf{1}}$;
  end
  $\boldsymbol{\mathcal{M}} \leftarrow \boldsymbol{K}$;
end

The proposal of local sharp matching through differentiable POT module reduces matching cost between the attention maps $\boldsymbol{\Phi}_{\mathcal{Y}}$ and $\boldsymbol{\Phi}_{\mathcal{X}}$. This is more effective way than the soft-pointer driven feature matching in DCP [42]. The resulting transportation or matching matrix $\boldsymbol{\mathcal{M}}$ from Algorithm 1 produces smaller number of matched features, but 'sharper' in probabilities matches compared to the soft-pointer approach of [42]. The first plot in Figure 3 with resolution[i] 102×102 explains that for any given source point $y_i \in \mathcal{Y}$ (along the column), its total matching probabilities with all other target points (along the row) are approximately constant. On the other hand, the other two plots in the same figure show how the same cost distribution, after optimizing $\boldsymbol{C}$ (setting $m = 0.1$ and $\xi = 5$ in Alg. 1), are optimally transported by $\boldsymbol{\mathcal{M}}$.

[i]Note that the original cost matrix $\boldsymbol{C}$ has dimension 1024×1024. We applied stride convolution with filter size 10 to lower its resolution for visual purpose. Hence the cost entries are only marginally scaled.

6DoF Pose Regression Loss. After estimating the optimal $\boldsymbol{\mathcal{M}}$, every point from the source point cloud $y_i \in \mathcal{Y}$ is mapped to the location

$$\tilde{y}_i \leftarrow \frac{1}{\sum_{j=1}^{N} \boldsymbol{\mathcal{M}}_{ij}} \sum_{j=1}^{N} \boldsymbol{\mathcal{M}}_{ij} y_i \tag{4}$$

that corresponds to its target position $\tilde{y}_i$. The loss function for sharp local feature matching-based pose estimation is a combination of pose loss $\mathcal{L}_{pose}$ and an auxiliary loss $\mathcal{L}_{aux}$ (as referred to by [8]). The pose loss is the ℓ_1-norm between the source points (*i.e.*, LiDAR points of current frame) transformed by the ground truth transformation $\mathbf{T}_{\text{gt}}$ and the predicted transformation $\mathbf{T}$:

$$\mathcal{L}_{pose} = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{T}_{\text{gt}} y_i - \mathbf{T} y_i| \text{ and } \mathcal{L}_{aux} = \frac{1}{N}\sum_{i=1}^{N} |\tilde{y}_i - \mathbf{T} y_i| \,. \tag{5}$$

The total odometry loss is defined as

$$\mathcal{L}_{odom} = \mathcal{L}_{pose} + \lambda_{aux}\mathcal{L}_{aux} \text{ where } \lambda_{aux} = 0.05. \tag{6}$$

3.2. PU-EPE: Evidential Pose Estimation

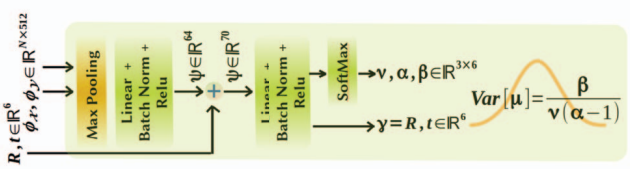


Figure 4. The PU-EPE network takes the embedding vectors $\phi_{\mathcal{Y}}$ and $\phi_{\mathcal{X}}$ of source and target frames, and predicts both aleatoric and epistemic uncertainty parameters [4].

Once optimal correspondence association is established, measurement noise is reduced. Though, out-of-order (OOD) data distribution can still induces uncertainties in

the next steps. To overcome this problem, aggressive data augmentation can be one option for a network to learn such OOD. While the POT-LDM can inherently learn the data uncertainty (*i.e.*, aleatoric uncertainty), it cannot automatically learn the model's predictive uncertainty (*i.e.*, epistemic uncertainty) [21]. The PU-EPE network, as shown in Figure 4, is trained jointly with the POT-LDM network to estimate the neural model's confidence in the LO predictions at different frames, *i.e.*, $\mathbf{T}_f, \mathbf{T}_{f+o}, \mathbf{T}_{f+2o}$, .., when frame-gap is o. In theory, the PU-EPE model learns to maximize the negative log-likelihood of the observed transformation values $\mathbf{T}_f \in \mathbb{R}^6$ that are assumed to be drawn from an independent and identically distributed (i.i.d) realization of a Gaussian distribution with unknown mean μ and variance σ^2. To learn the epistemic uncertainty parameters, the actual distribution mean μ and variance σ^2 are estimated by inferring the hyper-parameters γ, ν, α and β of a Normal-Inverse-Gamma (NIG) distribution $p(\mu, \sigma^2|\gamma, \nu, \alpha, \beta)$

$$= \frac{\beta^{\alpha}\sqrt{\nu}}{\Gamma(\alpha)\sqrt{2\pi\sigma^2}}(\frac{1}{\sigma^2})^{\alpha+1} exp\{-\frac{2\beta + \nu(\gamma-\mu)^2}{2\sigma^2}\} \quad (7)$$

as a posterior distribution[4] of the NIG.

By drawing i.i.d samples from the above NIG distribution, the PU-EPE model can directly infer both the epistemic uncertainty as $\text{Var}[\mu] = \frac{\beta}{\nu(\alpha-1)}$ and the aleatoric uncertainty as $\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha-1}$, and $\mathbb{E}[\mu] = \gamma$, without undertaking costly sampling [34] techniques. To infer the hyper-parameters of the NIG distribution, the transformation $\mathbf{T} \in \mathbb{R}^6$ predicted by the POT-LDM network is concatenated with the output of a linear feed-forward multi-layer perceptron (MLP) applied on $\phi_{\mathcal{Y}}$ and $\phi_{\mathcal{X}}$. Finally, the hyper-parameters are inferred using a second MLP as shown by green block 'Linear + Batch Norm. + ReLu'. Following Amini *et al.* [4], we define two losses namely, the negative log likelihood loss

$$\mathcal{L}^k_{NLL} = \frac{1}{2}log(\frac{\pi}{\nu_k}) - \alpha_k \log(\Omega_k)$$
$$+(\alpha_k + \frac{1}{2})\log((\mathbf{T}_k - \gamma_k)^2\nu_k + \Omega_k) + log(\frac{\Gamma(\alpha_k)}{\Gamma(\alpha_k + 1/2)}) \quad (8)$$

that minimizes the evidence on transformation errors, and the regularization loss

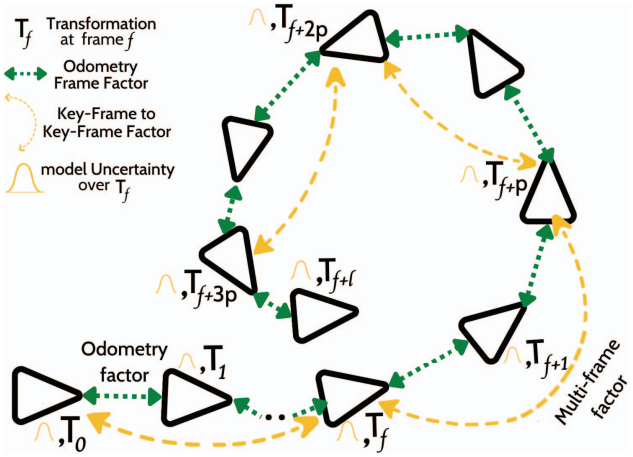
$$\mathcal{L}^k_R = |\mathbf{T}_k - \gamma_k| * (2\alpha_k + \nu_k) \quad (9)$$

that maximizes the model fitting, where $\Omega_k = 2\beta_k(1+\nu_k)$, and the subscript k for all hyper-parameters $\gamma_k, \nu_k, \alpha_k, \beta_k$ indicates the k^{th} element out of 6 (3 for Euler angles and 3 for translation) DoF. The total evidence loss

$$\mathcal{L}_{evidence} = \frac{1}{6}\sum_{k=1}^{6}(\mathcal{L}^k_{NLL} + \lambda_R \mathcal{L}^k_R) \quad (10)$$

is mean loss over all 6 transformation parameters. The regularization parameter λ_R is set to 0.2 for both the rotation and the translation parameters. The uncertainty in the model's LO predictions is used in the later stages of our method.

3.3. Pose Refinement using Uncertainty as Evidence



In the final functional component of DELO, the odometry predictions from POT-LDM $[.., \mathbf{T}_f, \mathbf{T}_{f+o}, \mathbf{T}_{f+2o}, \ldots]$, model's or epistemic uncertainty measures $[.., \text{Var}[\mu]_f, ..]$ from PU-EPE for all frames f in a given sub-sequence $\mathcal{S}$ are all streamed in for pose-graph optimization. Two parallel threads can run two tasks separately on different frame factors. The figure above depicts how one key-frame to another key-frame factor p for pose-graph optimization, and odometry frame-factor o can be set. Similar to some conventional methods [37, 33], we also employ GTSAM [13] for pose-graph optimization by matching the LO inference with LiDAR sensor rate. The orange curve in the figure shows factor graphs [23] built over local frames and previous key-frames with *multi-frame factors*. The intermediate frames can be more in numbers and hierarchical as well [15]. On the other hand green edges indicate frame-to-frame pose graph only up to odometry frame-factor. Once our DELO network is trained, we build an empty pose-graph during trajectory inference stage by adding the predicted poses $[\mathbf{T}_f, \mathbf{T}_{f+p}, \mathbf{T}_{f+2p}, ...]$. The odometry factor o and multi-frame pose-graph factor p are set to 2 and 4 respectively. If confidence score (*i.e.*, $1 - \text{Var}[\mu]$) of DELO network for LO prediction between two key-frames is bounded by predefined thresholds θ_{min} and θ_{max}, *i.e.*,

$$\theta_{min} \leq 1 - \text{Var}[\mu]_{f \mapsto f+p} \leq \theta_{max}, \quad (11)$$

then the factor graph node is rejected. This means, the network is confident and there is no need to refine the previous poses present in the current pose-graph.

4. Experiments and Evaluations

In this section, we present a complete and comprehensive experimental evaluation of our proposed DELO method on KITTI LiDAR odometry dataset [18].

Method	Training														Testing or Inference								
	00		01		02		03		04		05		06		07		08		09		10		(07-10)
	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}	runtime
ICP† [7]	2.99	6.88	2.58	11.2	3.39	8.21	5.05	11.1	4.02	6.64	1.92	3.97	1.59	1.95	3.35	5.17	4.93	10.04	2.89	6.93	4.74	8.91	-NA-
ICP$_{\perp}$† [35]	1.73	3.80	2.58	13.53	2.74	9.00	1.63	2.72	2.58	2.96	1.08	2.29	1.00	1.77	1.42	1.55	2.14	4.42	1.71	3.95	2.60	6.13	-NA-
LOAM† [20]	6.25	15.99	0.93	3.43	3.68	9.40	9.91	18.19	4.57	9.59	4.10	9.16	4.63	8.91	6.76	10.87	5.77	12.72	4.30	8.10	8.79	12.67	-NA-
LONet†[24]	0.72	1.47	0.47	1.36	0.71	1.52	0.66	1.03	0.65	0.51	0.69	1.04	0.50	0.71	0.89	1.70	0.77	2.12	0.58	1.37	0.93	1.80	80.1ms
PWCLO^{8†} [41]	**0.42**	**0.78**	**0.23**	**0.67**	**0.41**	**0.86**	**0.44**	**0.76**	**0.40**	**0.37**	**0.27**	**0.45**	**0.22**	**0.27**	0.44	0.60	**0.55**	1.26	**0.35**	**0.79**	**0.62**	1.69	-NA-
ICP [7]	2.23	13.4	8.44	14.9	4.81	30.4	2.16	10.5	1.53	90.7	1.37	18.7	1.41	32.3	0.79	7.85	1.25	12.6	5.11	32.3	2.81	21.1	3.8s
ICP$_{\perp}$ [35]	3.59	8.23	8.81	18.5	4.09	12.8	5.5	12.0	5.21	16.2	1.79	3.6	2.05	3.97	5.43	7.9	6.2	12.4	3.7	10.4	4.2	9.7	10.2s
DCP [42]	27.8	58.9	31.3	96.6	30.6	66.6	45.5	74.9	74.2	97.8	23.6	49.7	36.7	83.9	24.9	44.9	29.0	58.7	31.2	62.0	37.1	81.1	32ms
DGR [11]	5.95	35.9	12.8	53.6	7.23	46.3	7.80	61.0	13.3	56.7	4.69	33.2	21.8	43.5	5.83	36.4	7.81	37.9	7.74	43.4	6.92	51.7	682 ms
RPSRNet [2]	1.30	2.39	1.15	2.83	0.97	2.61	1.69	5.53	2.64	4.68	1.29	3.38	2.66	7.81	3.59	4.99	0.75	2.07	0.97	2.30	2.85	5.88	**20.3**ms
LOAM [47]	34.9	86.5	12.7	98.7	31.1	95.78	22.8	92.2	1.37	97.2	35.9	83.6	33.1	83.7	61.6	88.1	33.2	87.6	31.8	93.1	28.6	98.1	121ms
PWCLO^{1} [41]	19.5	30.1	3.41	7.90	12.8	25.9	43.9	37.2	19.9	22.9	24.1	34.7	13.1	12.1	12.50	18.0	19.1	30.8	14.7	21.8	19.9	33.9	77.3ms
PWCLO^{2} [41]	2.28	3.41	1.01	3.02	1.66	3.83	2.32	1.81	1.45	2.04	1.46	2.02	0.98	1.32	1.96	2.26	1.47	3.21	1.36	2.31	2.22	5.80	82.2
PWCLO^{8} [41]	**0.43**	**0.89**	**0.42**	**1.11**	0.76	1.87	**0.92**	1.42	**0.94**	**1.15**	0.71	1.34	0.38	**0.60**	1.00	1.16	0.72	1.68	**0.46**	**0.88**	**0.71**	2.14	125ms
DELO	1.30	2.97	2.19	11.99	1.71	4.88	1.58	3.34	7.42	2.42	1.00	2.17	1.01	2.58	1.44	1.97	3.48	9.02	1.54	2.26	2.16	3.54	35ms
DELO+PUEPE	0.81	1.43	0.57	2.19	**0.52**	**1.48**	1.10	**1.38**	1.70	2.45	**0.64**	**1.27**	**0.35**	0.83	**0.41**	**0.58**	**0.64**	**1.36**	0.57	1.23	0.90	**1.53**	41ms

Table 1. Results of different approaches for LiDAR odometry on KITTI [18] dataset are quantified by RRE, RTE metrics. The sequences **07-10** that are used for testing or inference, are '*not seen*' during training the network of the supervised approaches [24, 41, 42, 2] and ours. **Black**/Gray: The best and second best entries are underlined and marked in bold **black** and gray color.
† : Denotes the error metrics are reported from from [41]
PWCLO^{x}: The superscript x means ($x \times 1024$) number of input points are used for [41].

4.1. Dataset, Baselines, and Evaluation Metrics

Dataset. We randomly select 70% and 30% of frame ids from the sequences 00-06 as source frames f. These frames are independent and mutually exclusive from each other. Therefore, we set the corresponding target frames with ids $f + o$. The ground truth transformations $\mathbf{T}_{gt}^{f} = \mathbf{T}_{f+o}^{-1}\mathbf{T}_{f}$ for every frame f is set to the Velodyne coordinate system.

Baselines. We evaluate our proposed approach against standout baseline methods – PWCLO [41] network, LONet [24], RPSRNet[2], DGR [11], DCP [42], LOAM [48] without mapping (*i.e.*, no further scan-to-map alignment), ICP [7], and ICP$_{\perp}$ [35]. For evaluation, DCP, DGR, PWCLO network, and RPSRNet are re-trained. The POT-LDM and PU-EPE networks of DELO are jointly trained for 75 epochs with 1024 randomly selected points per scan, batch size of 16, and learning rate of 10^{-4} on two NVIDIA GeForce 1080Ti GPUs.

Evaluation Metrics. The angular deviation φ between the ground truth and the predicted rotations $(\mathbf{R}_{gt}, \mathbf{R})$ and the relative distance error Δt between the translation components $(\mathbf{t}_{gt}, \mathbf{t})$

$$\varphi = \frac{180^{\circ}}{\pi} \cos^{-1}\left(0.5(\text{tr}\left(\mathbf{R}_{gt}^{T}\mathbf{R}\right) - 1)\right), \text{ and } \Delta t = \|\mathbf{t}_{gt} - \mathbf{t}\| \tag{12}$$

quantify registration errors. On the other hand, the standard metrics to compare LO drifts are average of relative translation errors (RTE) and relative rotational errors (RRE) over all possible frames within the path lengths 100, 200, ..., 800 m: RTE t_{rel} in $\frac{\text{m}}{100\,\text{m}}$ as percentage %, RRE r_{rel} in $\frac{\text{degree}}{100\,\text{m}}$.

4.2. LiDAR Odometry Evaluation on KITTI

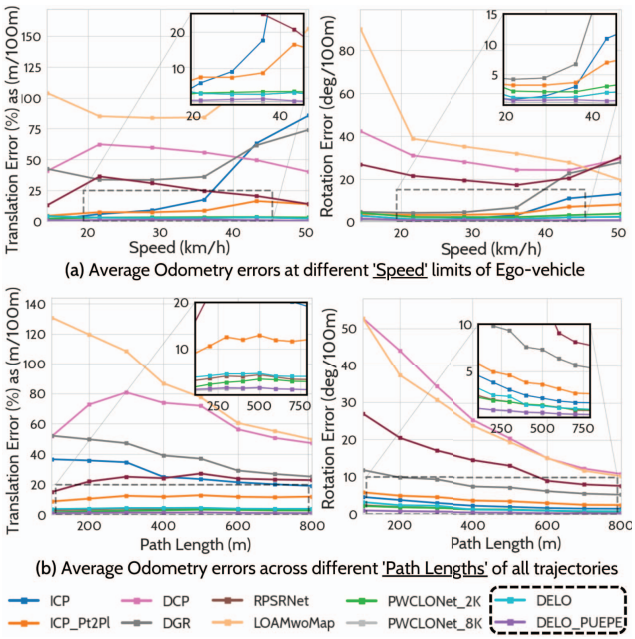


Figure 5. Relative transformation errors RRE and RTE averaged over all KITTI Odometry train and test sequences at different ranges of vehicle-speed (20Km/h, 30Km/h, ..., 50Km/h) and trajectory length (100m, 200m,...,800m). Our DELO+PUEPE performs the best.

Table 1 quantifies the most expressive error metrics [41, 24, 25], *i.e.*, RRE and RTE, for evaluating LO methods on every KITTI sequence. Lower values of the error tuple

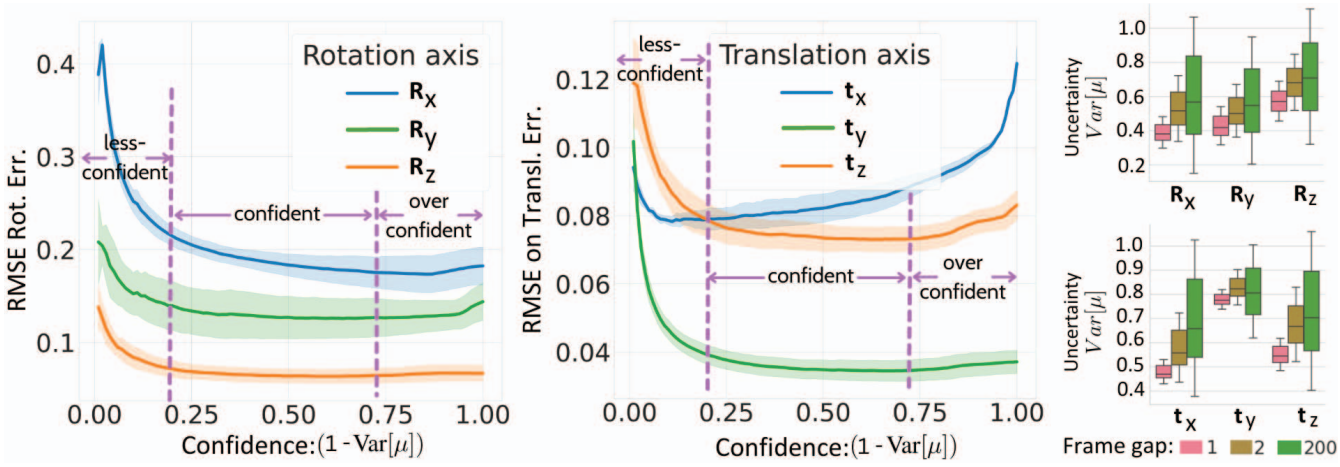


Figure 6. *On the left:* Root mean squared error (RMSE) along the transformation axes ($\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z, \mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z$) over all frames selected under confidence cut-off percentiles of our DELO+PUEPE network. *On the right:* Quartile plots over PU distribution along the transformation axes when different frame gaps are chosen for LO estimation.

($r_{\text{rel}}, t_{\text{rel}}$) indicate stability of a method for a prolonged duration of continuous navigation. Therefore, the higher values of ($r_{\text{rel}}, t_{\text{rel}}$) tuple reflect higher odometry errors due to drift accumulation along the trajectories. For a fair and consistent comparison, we split the training (including validation) and testing (or inference) set as Seq. **00-06** and **07-10** as per [41, 24]. While training our network, we optimize the combined losses defined in Eq. (6) and (10). To our understanding, LONet [24] and PWCLONet [41] are the two main benchmark methods to compete for performance superiority. Since the source codes LONet is not publicly available, we report its odometry errors (incl. ICP [7], ICP$_\perp$ [35], LOAM [47], and PWCLONet [41]) from [41] in separate rows of Table 1. In another separate part, reported errors are from our experiments.

After evaluation, the first major observation is the drop in accuracy of the deep learning-based methods on the test sequences compared to the training sequences (last three rows of Table 1). When there are only a handful of methods for deep learning-based LO, then the lack of their network's generalization ability on '*unseen*' data is a serious point to address. The proposed DELO, when trained with PU-EPE component, outperforms PWCLONet on three out of four test sequences. Although, on training sequences, PWCLONet outperforms our method only by marginal differences. Compared to the input size of 8K points in [41], DELO can accurately predict relative poses using matching correspondences between inputs of size 1K points per frame (see Fig. 3). Furthermore, if PWCLONet is trained with randomly sampled 2K and 1K points as input, we observe a significant jump in its odometry errors across all sequences (see Table 1). In terms of runtime, after exceptionally fast RPSRNet and DCP, our method takes ∼30-41ms that can match scanning frequency of any modern LiDAR sensor.

The next major observation is that state-of-the-art LiDAR data registration methods (*e.g.*, DCP [42], DGR [11], RPSRNet [2]) are not necessarily suitable for odometry task. For instance, on the test sequences, these methods score at least five times higher relative transformation errors than LONet [24], PWCLONet [41], and DELO+PUEPE (see Table 1 and Figure 5-(b)). When the ego-vehicle changes its speed or drifts during navigation, relative motion between selected source and target frames appear out-of-order than the learned distribution of sensor motion. The Figure. 5-(a) plots the relative transformation errors averaged over all possible frames from train and test sequences at different ranges of vehicle speed. The plots show our DELO, particularly DELO+PUEPE, and PWCLO-Net[8] methods are resistant to OOD. In contrast, other methods struggle to estimate correct relative motion when the ego-vehicle accelerates or decelerates.

Next, the performance of both point-to-point and point-to-plane ICP methods [7, 35] is derailed by continuous and catastrophic failures at different time steps. Similarly, methods like LOAM [47] or Lego-LOAM [20], well-known for distorted LiDAR-SLAM, overly rely on extra scan-to-map alignment step.

4.3. Equivariant PU-EPE as Evidence

In the final part of our analysis, we demonstrate approximately equivariant nature of the epistemic uncertainty $Var[\mu] = \frac{\beta}{\nu(\alpha-1)}$ w.r.t the odometry errors along all 6 directions (DoF) of the transformation parameters. The first two plots in Figure 6 show the RMSE on three Euler angles $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ for rotations, and three translation components $\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z$ over all frames selected under the different cut-off confidence values (as opposite of uncertainty, *i.e.* $1 - \text{Var}[\mu]$). The transparent width of each line denotes

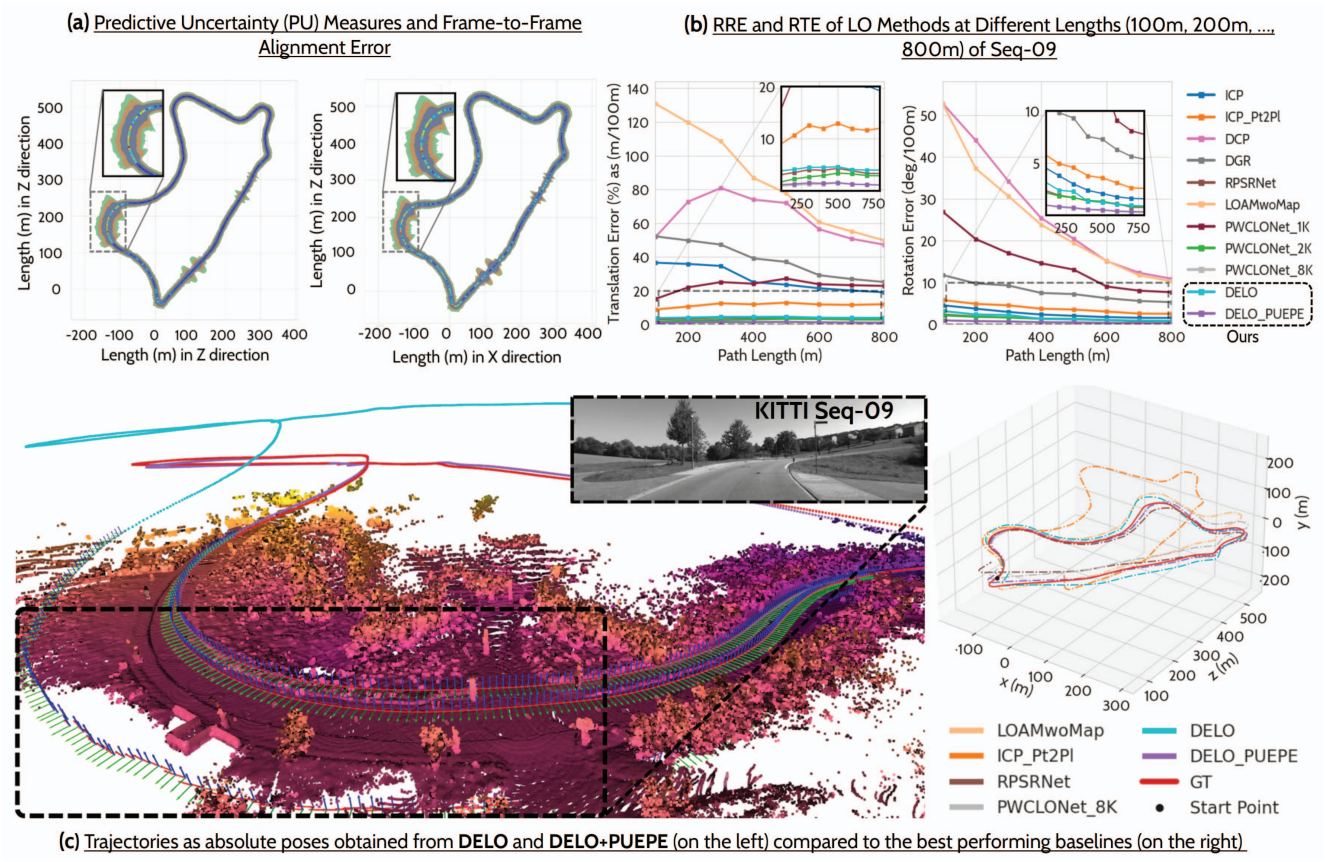


Figure 7. Complete experimental analysis on KITTI test Seq-09:

how much the RMSE values vary by increasing odometry frame gap o from 1 to 2. The last plot Figure 6 shows our model's uncertainty corresponding to different components of transformation axes, if different frame gaps (*i.e.*, 1, 2, and 200) are chosen to predict odometry. The confidence percentiles and RMSE over all frames, selected by every cut-off percentile, intuitively classify three regions where – DELO model is under-confident, confident, and over-confident. We set $\theta_{min} = 0.2$ and $\theta_{max} = 0.7$ in Eq. (11) using empirical analysis shown in Figure 6.

Finally, Figure 7 explains overall performance of our method using both the qualitative and quantitative results on the test sequence 09. Interestingly, the sequences 09 and 10 capture the same area via different routes, with narrow lanes covered by trees or bushes (see both the Figure 7-(c) and 1-(b)). We plot the trajectory color-mapped by the POT-LDM prediction errors φ and Δt (see Eq. 12), and the uncertainty values for $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z, \mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z$ as smooth 1D Gaussian filters along the same trajectory. It is noticeable in the Figure 7-(a), that there are clear evidences of continuous LO failures along the '*circular turning point*' where transformation errors are high and DELO+PUEPE network signals either its over-confidence or under-confidence. After online pose-refinement, our method recovers the accurate absolute poses of ego-vehicle and performs the best among baseline approaches (See the RRE, RTE errors and 3D trajectory plots in Figure 7-(b) and (c)).

5. Conclusions

This paper presents a real-time and LiDAR-only deep learning model for odometry estimation that jointly learns relative sensor motion and its predictive uncertainty. Our novel partial optimal transport network can learn sharp correspondence matching between two aggressively subsampled and non-uniformly distributed point clouds. The joint learning of odometry and its uncertainty leverages online pose-refinement by understanding under-confident or over-confident nature of predicted ego-motion. We show the equi-variant nature of PU across all transformation axes and all driving sequences can determine the thresholds for network to decide where sequential pose refinements are necessary. DELO is better than state-of-the-art methods, with great generalization ability, on KITTI dataset.

Acknowledgement. This work was partially funded by the project DECODE (01IW21001) of the German Federal Ministry of Education and Research (BMBF) and by the Luxembourg National Research Fund (FNR) under the project reference C21/IS/15965298/ELITE/Aouada.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. 3

[2] Sk A. Ali, K. Kahraman, G. Reis, and D. Stricker. RPSRNet: End-to-end trainable rigid point set registration network using barnes-hut 2^d-tree representation. In *CVPR*, 2021. 1, 2, 6, 7

[3] Sk A. Ali, K. Kahraman, D. Stricker, C. Theobalt, and V. Golyanik. Fast gravitational approach for rigid point set registration with ordinary differential equations. *IEEE Access*, 9:79060–79079, 2021. 1, 2

[4] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *NIPS*, 2020. 3, 4, 5

[5] Noam Shazeer Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polo-sukhin. Attention is all you need. In *NIPS*, 2017. 2, 3

[6] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyre. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2), 2015. 3

[7] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *TPAMI*, 14(2):239–256, 1992. 2, 6, 7

[8] Daniele Cattaneo, Matteo Vaghi, and Abhinav Valada. Lcdnet deep loop closure detection and point cloud registration for lidar slam. *IEEE Transactions on Robotics*, 2022. 3, 4

[9] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *IROS*, 2019. 1

[10] Younggun Cho, Giseop Kim, and Ayoung Kim. Unsupervised geometry-aware deep lidar odometry. In *ICRA*, 2020. 2

[11] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020. 1, 2, 3, 6, 7

[12] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NIPS*, 26, 2013. 1, 3

[13] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012. 5

[14] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. In *ICRA*, 2022. 2

[15] David Droeschel and Sven Behnke. Efficient continuous-time slam for 3d lidar-based online mapping. In *ICRA*, 2018. 5

[16] Olivier D Faugeras and Steve Maybank. Motion from point matches: multiplicity of solutions. *IJCV*, 4:225–246, 1990. 1, 2, 3

[17] Remi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurèlie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Lèo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 1, 3

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 5, 6

[19] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 2, 3

[20] Xingliang Ji, Lin Zuo, Changhua Zhang, and Yu Liu. Lloam: Lidar odometry and mapping with loop-closure detection based correction. In *International Conference on Mechatronics and Automation (ICMA)*, 2019. 2, 6, 7

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? NIPS, 2017. 3, 5

[22] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Globally consistent 3d lidar mapping with gpu-accelerated gicp matching cost factors. *IEEE Robotics and Automation Letters*, 6:8591 – 8598, 2021. 1

[23] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. 5

[24] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. Lo-net: Deep real-time lidar odometry. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 7

[25] Zhichao Li and Naiyan Wang. DMLO: Deep Matching Lidar Odometry. In *IROS*, 2020. 2, 6

[26] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. *CVPR*, 2019. 2

[27] Zhijian Liu, Alexander Amini, Sibo Zhu, Sertac Karaman Song Han, and Daniela L Rus. Efficient and robust lidar-based end-to-end navigation. In *ICRA*, 2021. 1, 3

[28] Zheng Liu and Fu Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021. 2

[29] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *ICCV*, 2019. 1, 2

[30] A. Myronenko and X. Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, 2010. 2

[31] Julian Nubert, Shehryar Khattak, and Marco Hutter. Self-supervised learning of lidar odometry for robotic applications. In *ICRA*, 2021. 2

[32] John Phillips, Julieta Martinez, Ioan Andrei Bârsan, Sergio Casas, Abbas Sadat, and Raquel Urtasun. Deep multi-task learning for joint localization, perception, and prediction. In *International Conference on Computer Vision and Pattern Recognition*, pages 4679–4689, 2021. 1, 3

[33] Milad Ramezani, Georgi Tinchev, Egor Iuganov, and Maurice Fallon. Online lidar-slam for legged robots with robust registration and deep-learned loop closure. In *ICRA*, 2020. 5

[34] Blasone Roberta-Serena, Vrugt Jasper A, Madsen Henrik, Rosbjerg Dan, Robinson Bruce A, and Zyvoloski George A. Generalized likelihood uncertainty estimation (glue) using adaptive markov chain monte carlo sampling. *Advances in Water Resources*, 31(4):630–648, 2008. 3, 5

[35] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435, 2009. 2, 6, 7

[36] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *NIPS*, 31, 2018. 3

[37] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IROS*, 2018. 1, 5

[38] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IROS*, 2018. 2

[39] V.Golyanik, Sk A. Ali, and D. Stricker. Gravitational approach for point set registration. In *CVPR*, 2016. 2

[40] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. 1, 3

[41] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. PWCLO-Net: Deep LiDAR Odometry in 3D Point Clouds Using Hierarchical Embedding Mask Optimization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 6, 7

[42] Yue Wang and Justin Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019. 1, 2, 3, 4, 6, 7

[43] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *NIPS*, 32, 2019. 2

[44] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 38(5):1–12, 2019. 3

[45] Ulrich Weiss and Peter Biber. Plant detection and mapping for agricultural robots using a 3d lidar sensor. *Robotics and autonomous systems*, 59(5):265–273, 2011. 1

[46] Z. J. Yew and G. H. Lee. Rpm-net: Robust point matching using learned features. In *CVPR*, 2020. 3

[47] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9, 2014. 2, 3, 6, 7

[48] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401 – 416, 2016. 1, 6

[49] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision (ECCV)*, 2016. 2