

Towards Robust LiDAR Localization: Deep Learning-based Uncertainty Estimation

Minoo Dolatabadi^{1,†}, Fardin Ayar^{1,†}, Ehsan Javanmardi², Manabu Tsukada², Mahdi Javanmardi^{1,*}

Abstract—LiDAR-based localization and SLAM often rely on iterative matching algorithms, particularly the Iterative Closest Point (ICP) algorithm, to align sensor data with pre-existing maps or previous scans. However, ICP is prone to errors in featureless environments and dynamic scenes, leading to inaccurate pose estimation. Accurately predicting the uncertainty associated with ICP is crucial for robust state estimation but remains challenging, as existing approaches often rely on handcrafted models or simplified assumptions. Moreover, a few deep learning-based methods for localizability estimation either depend on a pre-built map—which may not always be available—or provide a binary classification of localizable versus non-localizable, which fails to properly model uncertainty. In this work, we propose a data-driven framework that leverages deep learning to estimate the registration error covariance of ICP before matching, even in the absence of a reference map. By associating each LiDAR scan with a reliable 6-DoF error covariance estimate, our method enables seamless integration of ICP within Kalman filtering, enhancing localization accuracy and robustness. Extensive experiments on the KITTI dataset demonstrate the effectiveness of our approach, showing that it accurately predicts covariance and, when applied to localization using a pre-built map or SLAM, reduces localization errors and improves robustness.

I. INTRODUCTION

In recent years, autonomous vehicles have become an integral component of intelligent transportation systems, driving continuous research to push the boundaries of their capabilities. A key requirement for autonomous driving is achieving precise self-localization at the centimeter level [1]. The Global Navigation Satellite System (GNSS) is a cost-effective and widely used method for vehicle localization. Although GNSS delivers reliable positioning in open-sky environments, its accuracy is significantly compromised in urban settings due to factors such as signal blockage, non-line-of-sight (NLOS) conditions, and multipath effects [2]. To address these challenges, vision-based approaches—particularly those leveraging LiDAR—have been proposed as alternative or complementary solutions [3]. In these systems, map matching techniques like the Iterative Closest Point (ICP) algorithm are frequently employed to align sensor data with pre-existing maps, thereby enhancing localization precision [4].

However, vision-based localization methods—like other relative localization techniques—are susceptible to error ac-

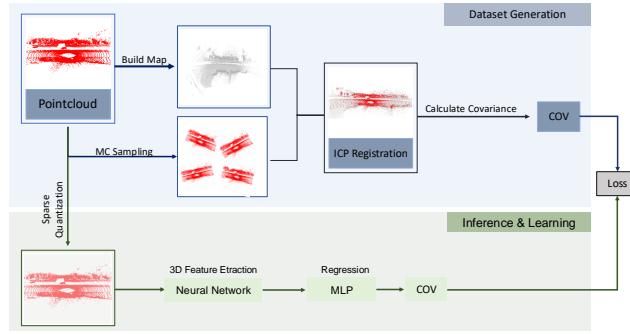


Fig. 1. Overview of the proposed approach. In the upper section, a map is built from the input point cloud, followed by ICP matching to generate covariance for each point cloud. In the lower section, the input point cloud from the LiDAR sensor is processed through a neural network. The entire network, including feature extraction and regression, is trained end-to-end.

cumulation, where minor errors can progressively lead to significant drift over time. This phenomenon is observed in both Simultaneous Localization and Mapping (SLAM) and pre-built map-based localization approaches, although it tends to be more pronounced in SLAM [5]. In the context of ICP, previous work has demonstrated that factors such as featureless environments (e.g., tunnels) and the presence of dynamic objects can adversely affect the matching accuracy [6].

State estimation methods, such as Kalman filtering, are commonly employed to mitigate these problems. Yet, these techniques depend on an accurate error model (often represented in the simplest form by an error covariance matrix), which is challenging to determine for matching algorithms [7].

To address this limitation, several studies have proposed data-driven approaches to predict ICP error, either using classification-based methods [8] or by directly estimating the covariance [7].

Specifically, we introduce a data-driven framework that predicts the full six-degree-of-freedom (6-DoF) ICP registration covariance from a *single* LiDAR scan *prior* to correspondence search and ICP refinement. The output is a symmetric positive-definite (SPD) 6×6 covariance on $SE(3)$ suitable for probabilistic fusion. An overview of the pipeline is shown in Fig. 1.

Contributions.

- **Pre-ICP, per-scan $SE(3)$ covariance.** We predict the full 6×6 registration covariance directly from a single LiDAR scan, capturing translation–rotation coupling

[†] Minoo Dolatabadi and Fardin Ayar contributed equally to this work.

¹Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.

²Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan.

*Corresponding Author: Mahdi Javanmardi, mjavan@aut.ac.ir.

before ICP is run. Per-scan training targets are obtained by estimating empirical covariances from Monte Carlo ICP under randomized initializations.

- **Kalman fusion and evaluation.** Each scan is paired with a reliable 6-DoF covariance, used as the measurement noise in a standard Kalman filter. This improves pose accuracy over fixed or heuristic covariances on the evaluated sequences.
- **Practicality.** The model does not require a pre-built map at inference, supporting both SLAM and map-based operation.

II. RELATED WORKS

As noted earlier, LiDAR-based localization typically registers scans with ICP [9][10]; however, ICP can be unreliable in feature-poor or geometrically degenerate scenes [6]. Predicting how well a LiDAR scan can be aligned before or during ICP is a crucial first step toward robust localization. Existing research in this area falls broadly into two complementary groups. i) *Analytic Methods*, which use geometric and optimization diagnostics (e.g., scene structure, Jacobian conditioning, eigenvalue spectra) to detect degeneracy, and (ii) *Learning-Based Methods*, which replace fixed heuristics with data-driven models that infer reliability and uncertainty directly from raw sensor data—yielding scalar or binary risk/localizability scores and learned covariance predictions (full 6×6 pose uncertainty). This trajectory reflects a shift from explicit, model-based indicators to learned predictors of both risk and covariance; we review both in turn.

A. Analytic Methods

Classical approaches detect ICP failure modes by analyzing the registration cost and its local curvature. In practice, two strategies are common: (i) geometric or observability checks—e.g., looking for large planar areas or symmetric structures in the point cloud to flag ICP degeneracy [11][12]—and (ii) curvature-based tests that inspect the Hessian’s eigenvalues or the ICP Jacobian’s condition number to reveal poorly constrained directions[13][14]. While these techniques improve alignment stability, they typically require both the source and target scans and rely on heuristic thresholds, thereby limiting robustness in dynamic or sparse scenes. More recent variants address degeneracy like X-ICP [6] normalizes constraints to expose rank loss, and DARE-SLAM [14] handles perceptual aliasing; however, these improvements are applied after the fact rather than predicting issues upfront.

B. Learning-Based Methods

To overcome the above limitations, researchers have explored data-driven predictors of localization before or alongside alignment, as well as learned models of registration uncertainty. Prior work divides into two complementary strands. The first predicts or detects alignment risk—often before or during ICP—by measuring scan overlap or scene “localizability”, yielding a scalar or binary failure indicator. The second estimates the full 6×6 pose covariance of the

ICP result, providing probabilistic uncertainty for Kalman filtering and other sensor-fusion methods. In practice, risk detection helps prevent catastrophic misalignments, while covariance estimation supplies the detailed uncertainty needed for consistent downstream state estimation. We review these strands in turn.

1) *Risk/Localizability Prediction:* Some learning-based methods predict pre-alignment localizability (risk) directly from raw LiDAR scans, enabling early detection of ICP failures. Nobili et al. [15] train an SVM classifier on scan overlap and LiDAR intensity features to predict alignment failure probabilities. Similarly, Adolfsson et al. [16] quantify entropy change between scans to detect poor alignments. These methods produce binary or scalar indicators of ICP reliability rather than a full uncertainty measure. More recent deep learning approaches go further: Chen et al. [17] propose OverlapNet, a Siamese network estimating overlap between LiDAR scans , effectively predicting *matchability*. SuperLoc [18] integrates LiDAR and inertial data with a neural alignment risk predictor, enabling early detection of localization failures in feature-scarce environments. Nubert et al. [8] present a learning-based localizability estimator that outputs a degeneracy score for LiDAR localization. In parallel, multisensor frameworks have been developed to mitigate degeneracy on the fly. For instance, Xie et al. [19] introduce a Selective Kalman Filter that fuses camera and LiDAR data only when the LiDAR-only SLAM is predicted to be degenerate. By adaptively injecting visual constraints when needed, their system improves robustness during ICP failure cases.

In addition to ICP-based approaches, NDT-based localization has been systematically studied. Javanmardi et al. [20] examined factors such as feature completeness and environmental structure to predict localization errors, demonstrating that map accuracy alone is not enough for precise localization. Their analysis successfully modeled urban scene errors based on these map features. Endo et al. [21] further explored how dynamic elements, such as moving vehicles, influence localization accuracy, providing insights into improving NDT-based localization in real-world environments. These advancements in both ICP and NDT-based systems highlight the transformative potential of data-driven methods in enhancing alignment and localization accuracy.

2) *Covariance Prediction:* Learning covariance with deep networks is challenging: the estimator must output a valid symmetric positive semidefinite matrix that captures both marginal scales and cross-correlations, while remaining numerically stable for real-time use. In related estimation tasks, Liu et al. [22] introduced a neural network framework that predicts sensor measurement covariance, offering real-time adaptability. Additionally, Fang et al. [23] applied deep learning to predict covariance matrices for position estimation, ensuring semi-positive definiteness using Cholesky decomposition. In ICP-based localization, early work by Censi derived a closed-form covariance expression for ICP, laying the foundation for uncertainty analysis but relying on small-noise assumptions and linearization around the converged

solution [24]. Building on this idea, CELLO and CELLO-3D propagate geometric error sources to compute registration covariance in real time [25], [7]; however, like other analytic or feature-engineered methods, they still produce covariance only *after* alignment and can be sensitive to poor initialization or scene structure. More recently, De Maio and Lacroix learn a Bayesian model of the ICP *alignment error after registration*, yielding a data-driven, post-hoc covariance for the final solution [26].

Discussion. We address the missing *pre-ICP* regime by estimating a full 6×6 SE(3) covariance *before* any correspondence search, from a single LiDAR scan—map-free and threshold-free. In contrast to methods that output only binary/scalar risk cues or rely on external degeneracy detectors [8], [19], our model infers degeneracy directly from the scan and returns a continuous, symmetric positive-definite covariance that captures both scale and directionality, making it directly usable in probabilistic filters. Post-registration estimators (e.g., [26]) recover uncertainty *after* alignment; our one-shot, data-driven predictor provides it *a priori*. We train on Monte-Carlo-aligned covariance targets using modern point-cloud backbones (PointNet++, Cylinder3D), and analyses of ICP convergence behavior.

III. PROPOSED APPROACH

Given a 3D LiDAR point cloud scan $s_k \in \mathbb{R}^{n_k \times 3}$, our objective is to predict a 6×6 covariance matrix Σ_k that quantifies the uncertainty in the estimated relative pose if this point was matched against a reference map. In this work, we address two scenarios: one in which a pre-built map is available, and another where a local map generated via SLAM is used.

The diagonal elements of the covariance matrix are denoted by

$$\mathbf{c}_k = \text{diag}(\Sigma_k) = (\sigma_{t_x}^2, \sigma_{t_y}^2, \sigma_{t_z}^2, \sigma_{\omega_x}^2, \sigma_{\omega_y}^2, \sigma_{\omega_z}^2)^\top,$$

where the first three entries are translational variances (m^2) and the last three are rotational variances (rad^2 or degree 2) about the $x/y/z$ axes.

our method is built upon two key components:

- 1) **Monte Carlo-based Dataset Generation:** This component creates a dataset from pre-built maps by generating ground truth 6×6 error covariance matrices, which capture the uncertainty across various localization dimensions.
- 2) **Deep Neural Network Framework:** This component is designed and trained to predict the error covariance matrix from the LiDAR scans. The predicted covariance, with its interpretable diagonal \mathbf{c}_k , directly informs on the uncertainty of the localization estimate.

An overview of the proposed method is presented in Figure 1.

A. ICP and Its Uncertainty

We utilize the Lie group $SE(3)$ for rigid body transformations, which combines rotation and translation in three-dimensional space. The transformation matrix $\mathbf{T} \in SE(3)$

can be expressed as:

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}, \quad (1)$$

where $\mathbf{R} \in SO(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is a translation vector. To handle transformations analytically, it's beneficial to map \mathbf{T} to its corresponding Lie algebra $\mathfrak{se}(3)$ using the logarithm map:

$$\boldsymbol{\tau} = \log(\mathbf{T}) \in \mathfrak{se}(3). \quad (2)$$

The vector $\boldsymbol{\tau} \in \mathbb{R}^6$ comprises the translational and rotational components:

$$\boldsymbol{\tau} = \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\omega} \end{pmatrix}, \quad (3)$$

where $\mathbf{u} \in \mathbb{R}^3$ represents translation, and $\boldsymbol{\omega} \in \mathbb{R}^3$ denotes the rotation in angle-axis form. This representation facilitates the expression of uncertainties in transformations as a covariance matrix $\Sigma \in \mathbb{R}^{6 \times 6}$:

$$\Sigma = \begin{pmatrix} \Sigma_{uu} & \Sigma_{u\omega} \\ \Sigma_{\omega u} & \Sigma_{\omega\omega} \end{pmatrix}, \quad (4)$$

where Σ_{uu} and $\Sigma_{\omega\omega}$ are the covariance matrices for translation and rotation, respectively, and $\Sigma_{u\omega}$ captures the correlation between them.

In practical applications, an initial transformation \mathbf{T} serves as a prior, often derived from odometry or other sensors. The Iterative Closest Point (ICP) algorithm refines this to obtain an estimated transformation $\hat{\mathbf{T}}$:

$$\hat{\mathbf{T}} = \text{ICP}(\mathbf{P}, \mathbf{Q}, \mathbf{T}), \quad (5)$$

where \mathbf{P} and \mathbf{Q} are the source and target point clouds, respectively. The uncertainty associated with $\hat{\mathbf{T}}$ is modeled as:

$$\hat{\mathbf{T}} = \exp(\boldsymbol{\xi})\bar{\mathbf{T}}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \Sigma), \quad (6)$$

assuming the perturbation $\boldsymbol{\xi}$ follows a Gaussian distribution with zero mean and covariance Σ and $\bar{\mathbf{T}}$ being the ground truth transformations.

To summarize, our goal is to learn a function $f_\theta : s_k \in \mathbb{R}^{n_k \times 3} \rightarrow \Sigma_k \in \mathbb{R}^{6 \times 6}$, which is a deep neural network parameterized by θ that predicts an error covariance matrix of the ICP algorithm, given only a scanned LiDAR point cloud.

B. Dataset Generation

To train the deep neural network for predicting the error covariance matrix in localization, we use a dataset generation process based on Monte Carlo sampling like [8]. This process addresses the sensitivity of the ICP algorithm to its initial guess by repeatedly registering each point cloud under varying perturbations. Following the previous works [8], the overall procedure for each point cloud s_k is as follows:

1) Map Construction:

A map is constructed by aggregating a sequence of point clouds—specifically, x preceding and y subsequent scans. Each scan is transformed from its local coordinate frame into the map's coordinate system using

ground truth transformations. Notably, when estimating error covariance for SLAM, only preceding scans are used (i.e., $y = 0$). This sequential accumulation progressively constructs an accurate representation of the environment. In cases where ground truth data are unavailable, mapping methods such as LOAM [27] can be employed as an alternative.

2) Data Reduction:

To mitigate computational load and memory requirements, both the constructed map and the reference point cloud are downsampled via voxel filtering. In our implementation, the map is downsampled with a voxel size of 1 meter, while the reference point cloud is downsampled at a finer resolution of 0.1 meter.

3) ICP Matching:

For each $i \in \{0, \dots, N\}$, we create an initial transformation \mathbf{T} (see Eq. 5) assuming it follows the model:

$$\mathbf{T}_i = \exp(\boldsymbol{\xi}_o) \bar{\mathbf{T}}_k, \quad (7)$$

where $\boldsymbol{\xi}_o \sim \mathcal{N}(\mathbf{0}, \mathcal{O})$ represents a perturbation around the ground truth transformation $\bar{\mathbf{T}}_k$. The 6×6 covariance matrix \mathcal{O} is a predefined hyperparameter representing the real-world uncertainty of the initial transformation (see Sec. IV). The exponential map converts $\boldsymbol{\xi}_o$ into a valid transformation in $SE(3)$.

After applying the point-to-plane ICP, the refined transformation $\hat{\mathbf{T}}_i$ is obtained for the i -th sampled initial transformation.

4) Error Calculation and Covariance Estimation:

For each $\hat{\mathbf{T}}_i$, the registration error is quantified by:

$$\boldsymbol{\xi}_i = \log(\bar{\mathbf{T}}_k^{-1} \hat{\mathbf{T}}_i), \quad (8)$$

where the logarithm map transforms the error into its vector representation in $\mathfrak{se}(3)$. After repeating the registration process for N samples, the covariance matrix is estimated by:

$$\mathbf{Y}_k = \frac{1}{n-1} \sum_{i=1}^n \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\top, \quad (9)$$

where $\mathbf{Y}_k \in \mathbb{R}^{6 \times 6}$ represents the error covariance matrix for the k -th point cloud.

The complete pseudocode for the dataset generation process is provided in Algorithm 1.

C. Deep Learning for Predicting Error Covariance Matrix

Our goal is to optimize a deep regression network f_θ that predicts error covariance matrices from features extracted from the LiDAR scan s_k . In this section, we first describe the network architecture, followed by our data augmentation strategy, and finally the loss function formulation.

1) *Network Architecture*: We employ state-of-the-art architectures, Cylinder3D [28] and PointNet++ [29], to capture both local and global geometric characteristics. The extracted features are subsequently passed through a multi-layer perceptron (MLP) for regression. To guarantee that the predicted

Algorithm 1: Dataset-Generation

```

Input: Set of point clouds
Output: Covariance matrices  $\mathbf{Y}_k$  for all  $k$ 
1 for each point cloud  $s_k$  do
2   for  $i \leftarrow 1$  to  $n$  do
3     Construct map using  $x$  previous and  $y$ 
      subsequent point clouds
4     Transform point clouds from local to global
      coordinates using ground truth
5     Initialize  $\boldsymbol{\xi}_o \sim \mathcal{N}(\mathbf{0}, \mathcal{O})$ 
6     Compute  $\mathbf{T}_i = \exp(\boldsymbol{\xi}) \bar{\mathbf{T}}_k$ 
7     Apply ICP to align input point cloud with
      map, obtain estimate  $\hat{\mathbf{T}}_i$ 
8     Compute  $\boldsymbol{\xi}_i = \log(\bar{\mathbf{T}}_k^{-1} \hat{\mathbf{T}}_i)$ 
9     Compute covariance matrix
10     $\mathbf{Y}_k = \frac{1}{n-1} \sum_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i^\top$ 
11 return Covariance matrices  $\mathbf{Y}_k$  for all  $k$ 

```

covariance matrices are symmetric and positive definite, the network is designed to output a lower triangular matrix C_t . The final covariance prediction is then reconstructed as:

$$\hat{\mathbf{Y}} = C_t C_t^\top. \quad (10)$$

This formulation ensures that the predicted error covariance retains the necessary structural properties, namely, being positive semi-definite. We also add a small diagonal term $\epsilon \mathbf{I}$ to make it strictly positive definite and numerically stable.

2) *Sampling Strategy and Data Augmentation*: Due to the prevalence of samples with near-zero covariance in the dataset (Fig. 2), a weighted sampling strategy is employed during training. Specifically, the probability of selecting a sample for the input batch of our network is proportional to the magnitude of its largest covariance element, thereby emphasizing samples with higher error covariance. To further enhance model robustness, we apply a data augmentation strategy focused on rotational transformations. Each reference point cloud is randomly rotated about the vertical (z) axis to simulate various orientations. Following a rotation, the corresponding covariance matrices are adjusted using the adjoint representation of the transformation:

$$\bar{\mathbf{Y}} = \mathbf{A}_T \mathbf{Y} \mathbf{A}_T^\top, \quad (11)$$

where \mathbf{A}_T is the adjoint matrix associated with the applied rotation T . This augmentation increases the diversity of the training data while preserving the underlying geometric structure of the environment.

3) *Loss Function*: The training objective is formulated as a regression problem aimed at minimizing the difference between the predicted and ground truth covariance matrices.

TABLE I

THE PERTURBATION COVARIANCE PARAMETERS FOR GENERATING DATASETS

Parameter	σ_x [m]	σ_y [m]	σ_z [m]	σ_ϕ [$^\circ$]	σ_θ [$^\circ$]	σ_ψ [$^\circ$]
Value	1.0	1.0	0.2	5.0	5.0	10.0

The loss function is defined as:

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{Y}}, \bar{\mathbf{Y}}) &= \alpha D_{KL}\left[\mathcal{N}(0, \hat{\mathbf{Y}}) \parallel \mathcal{N}(0, \bar{\mathbf{Y}})\right] \\ &+ \beta \sum_{i=1}^n L_{\text{Huber}}(\hat{\mathbf{Y}} - \bar{\mathbf{Y}}), \end{aligned} \quad (12)$$

where $\alpha = 0.01$ and $\beta = 1.0$ are empirically determined hyperparameters, $\hat{\mathbf{Y}}$ is the predicted covariance matrix, and $\bar{\mathbf{Y}}$ is the ground truth. The Huber loss is defined as:

$$L_{\text{Huber}}(\delta) = \begin{cases} \frac{1}{2}\delta^2, & |\delta| \leq \delta_0, \\ \delta_0 \left(|\delta| - \frac{1}{2}\delta_0\right), & |\delta| > \delta_0. \end{cases} \quad (13)$$

Because the covariance matrix is symmetric, the Huber loss is applied only to its upper (or equivalently, lower) triangular elements.

IV. EXPERIMENT AND EVALUATION

A. Dataset

Our proposed method is designed to be generalizable and applicable to any LiDAR-based dataset for localization. In this work, however, we conduct our experiments using the KITTI Odometry dataset [30], which comprises 11 sequences with ground truth 6DoF poses. We extracted a point cloud every 50 scans (approximately every 5 seconds) from each sequence and followed Algorithm 1 to construct the dataset. For evaluation purposes, each sequence was partitioned into three consecutive segments with a split ratio of 70:20:10 for training, testing, and evaluation, respectively. For the creation of the reference map (e.g., x and y in Algorithm 1), we explored two scenarios:

- 1) **Scenario 1 (Pre-built maps):** Utilizes 10 preceding frames and 20 subsequent frames, emulating scenarios where a high-definition (HD) pre-built map is available for vehicle localization.
- 2) **Scenario 2 (SLAM):** Utilizes only 10 preceding frames, relying exclusively on historical data, making it appropriate for simultaneous localization and mapping applications.

The perturbation covariance matrix (ξ_o in Algorithm 1) is a critical component of our approach. The perturbation covariance matrix is parameterized by its diagonal elements, which represent the uncertainties of the initial transformations. These parameters are summarized in Table I.

In Fig. 2, the histograms depict the localization error variance across various dimensions for a sequence of the kitti dataset. As expected, most points show low error covariance, while the variance is most pronounced along the x-axis (indicating the vehicle's forward direction) and the y-axis (indicating the vehicle's leftward direction).

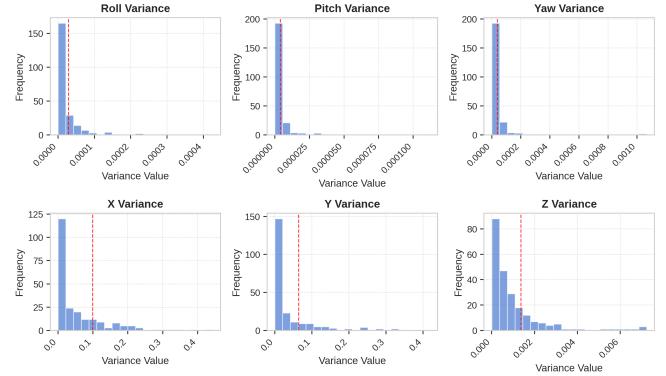


Fig. 2. Histogram of localization error variance for sequence 00 of the kitti dataset, generated with Algorithm 1. The red line indicates the 80% percentile.

TABLE II
EVALUATION METRICS FOR DIFFERENT NETWORK CONFIGURATIONS AND DATASET STRATEGIES

Network	KL	x (MAE)	y (MAE)	yaw (MAE)
Localization				
PointNet++	3.69	0.059	0.041	3.21×10^{-5}
Cylinder3D	4.06	0.117	0.071	5.50×10^{-5}
Cylinder3D (pretrained)	3.97	0.061	0.051	5.36×10^{-5}
SLAM				
PointNet++	4.28	0.063	0.035	2.55×10^{-5}
Cylinder3D	4.70	0.072	0.055	5.52×10^{-5}
Cylinder3D (pretrained)	4.32	0.084	0.053	5.441×10^{-5}

B. Results for Covariance Estimation

Table II presents the evaluation metrics for various network configurations and dataset strategies. The results indicate that PointNet++ outperforms the other models, particularly in the localization setting, where it achieves the most accurate covariance estimations with lower errors across all evaluated metrics. This suggests that its hierarchical feature extraction mechanism effectively captures relevant geometric information for uncertainty estimation. However, its performance slightly deteriorates in the SLAM setting, where the absence of subsequent frames impacts the estimation accuracy.

In contrast, Cylinder3D exhibits higher errors across all settings, suggesting that it is less effective in capturing precise covariance estimations. One possible explanation is that Cylinder3D's voxel-based representation may lead to information loss in the context of fine-grained error estimation, which is crucial for accurate pose uncertainty prediction. Notably, the pretrained Cylinder3D model—which was trained on a point cloud segmentation task—demonstrates an improvement over its standard counterpart. This improvement highlights the potential benefits of pretraining on structured point cloud tasks, yet the model still performs sub-optimally compared to PointNet++. These findings suggest that further research is required to explore more effective pretraining strategies that can better transfer domain-specific knowledge to error covariance estimation.

In addition to these results, we visualize the predicted co-

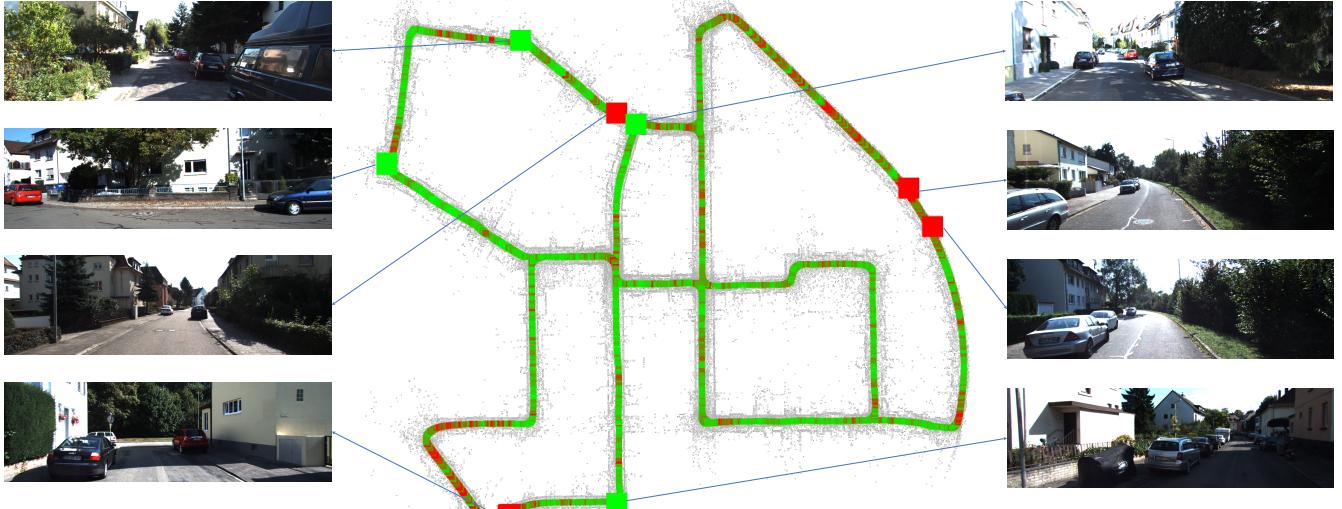


Fig. 3. The map is reconstructed from consecutive LiDAR scans in KITTI Odometry Sequence 00. The trajectory is color-coded by $\text{tr}(\Sigma_{xy}) = \sigma_x^2 + \sigma_y^2$ (green low, red high). As expected, we observe smaller error covariance at turns and crossroads, and larger covariance along straight segments.

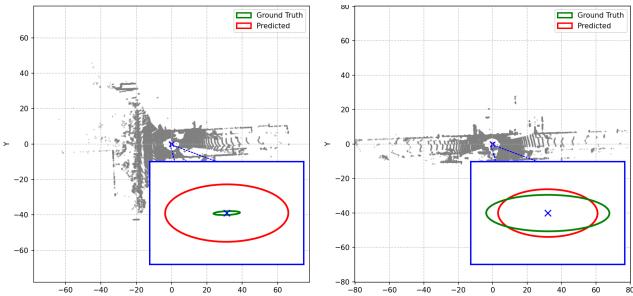


Fig. 4. Two different predication of our best model in comparison to GT error covariance in x and y directions.

variance along a single representative sequence of the KITTI dataset (Fig. 3). The map shows the vehicle’s trajectory, with each pose color-coded by the trace of the $x - y$ position-covariance submatrix, $\text{tr}(\Sigma_{xy}) = \sigma_x^2 + \sigma_y^2$ (green = low, red = high). Empirically, red segments appear on straight right-of-way portions of the route, whereas turns tend to appear green—likely because turning increases viewpoint diversity and geometric constraints (parallax), improving observability and lowering $\text{tr}(\Sigma_{xy})$. This visualization allows us to pinpoint exactly where the model’s performance degrades.

To further analyze the performance of our models, we visually examined the predicted covariance values by plotting two representative outputs in Fig. 4: one with a low KL divergence and one with a high KL divergence. Interestingly, our model tends to predict an uncorrelated covariance between the x and y directions, with generally higher values along the vehicle trajectory (x -direction). This observation is consistent with real-world scenarios, where uncertainty along the direction of motion is typically greater.

TABLE III
PER-SEQUENCE MEAN IMPROVEMENT (%) OF PER-FRAME COVARIANCE
OVER A FIXED BASELINE ON KITTI ODOMETRY.

Seq. #	Split	Avg APE Improvement (%)	Avg RPE Improvement (%)
00	val/test	0.52	2.17
01	val/test	10.60	1.92
02	val/test	3.64	4.61
04	val/test	2.71	1.45
05	val/test	0.02	3.79
06	val/test	0.64	1.74
07	val/test	0.71	9.26
08	val/test	-0.19	-0.02
09	val/test	0.49	2.81
10	val/test	1.13	0.13

C. Kalman–Filter Integration

We smooth raw ICP poses with a constant-acceleration EKF (translation) and constant-angular-acceleration prior (rotation), without inertial sensing. The nominal state maintains rotation $\mathbf{R}_k \in SO(3)$ and a 15-vector $[\mathbf{p}_k, \mathbf{v}_k, \mathbf{a}_k, \boldsymbol{\omega}_k, \dot{\boldsymbol{\omega}}_k]$.

Between consecutive scans, we propagate the state using the scan time step and the constant-acceleration/constant-angular-acceleration priors, yielding a pose prior \mathbf{T}_k^- . Each scan provides a 6-DoF ICP pose $\hat{\mathbf{T}}_k^{\text{ICP}}$. We form a pose residual on $SE(3)$ and apply a standard first-order EKF update around \mathbf{T}_k^- .

Our network outputs a per-scan covariance $\hat{\mathbf{Y}}_k \in \mathbb{R}^{6 \times 6}$ before matching. For temporal stabilization, we replace it with a 5-frame moving average $\tilde{\mathbf{Y}}_k$ (last five predictions), and use this as the measurement noise. As covariances are expressed in a local/sensor frame, we map them to the world frame using the $SE(3)$ adjoint before using.

Evaluation protocol. We compare our *per-frame* covariance weighting against a *fixed* covariance baseline on three KITTI

odometry sequences. For a fair comparison, the fixed baseline uses the *per-sequence mean* of our predicted covariances as its constant measurement noise, and all Kalman filter hyperparameters are tuned once on this baseline and then reused *unchanged* for our method. We operate in a pure LiDAR SLAM regime (no global sensors, no pose-graph optimization, no loop closure). Each sequence is partitioned into non-overlapping 200-frame windows; metrics are computed per window and then averaged. Because the first 70% of each sequence is used to train the covariance predictor, we report *val/test*, and present mean Absolute Pose Error (APE) % and Relative Pose Error (RPE) % improvements over the fixed baseline.

As shown in Table III, across the 10 KITTI Odometry sequences, our per-frame covariance generally improves or equals the fixed baseline on the held-out (val/test) splits for both APE and RPE. Training-split gains (not shown) are of similar magnitude, indicating the predictor is not overfitting to sequences it has seen. Overall, replacing a static noise model with our learned, scene-aware covariance yields cleaner trajectories without extra sensors or loop closure.

V. CONCLUSION

In this work, we introduced a deep learning-based framework for predicting the registration error covariance of the Iterative Closest Point (ICP) algorithm in LiDAR-based localization and SLAM systems. Our method directly estimates uncertainty from raw point clouds, improving sensor fusion through more accurate covariance modeling in Kalman filtering. Through extensive experiments on the KITTI-Odometry dataset, we demonstrated that our approach enhances localization accuracy. Our visual analysis further highlighted the model’s tendency to predict uncorrelated covariances along the vehicle trajectory, suggesting the need for a more diverse dataset to refine estimation accuracy. Moreover, integrating our predicted covariance into a Kalman filter demonstrated notable improvements in trajectory estimation. Our model outperformed a fixed covariance approach in both APE and RPE. Future research will focus on optimizing network architectures, exploring alternative pretraining methods, and increasing dataset diversity to improve model robustness. Additionally, real-time deployment and multimodal sensor fusion, incorporating data from cameras and radar, will be investigated to further enhance localization accuracy in complex environments.

REFERENCES

- [1] E. Javanmardi, Y. Gu, M. Javanmardi, and S. Kamijo, “Autonomous vehicle self-localization based on abstract map and multi-channel lidar in urban area,” *IATSS research*, vol. 43, no. 1, pp. 1–13, 2019.
- [2] C. Ellul, M. Adjrad, and P. Groves, “The impact of 3d data quality on improving gnss performance using city models-initial simulations,” in *ISPRS Annals*, vol. 4. International Society for Photogrammetry and Remote Sensing, 2016.
- [3] Y.-W. Seo and R. Rajkumar, “Tracking and estimation of ego-vehicle’s state for lateral localization,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 1251–1257.
- [4] A. Charroud, K. El Moutaouakil, V. Palade, A. Yahyaoui, U. Onyekpe, and E. U. Eyo, “Localization and mapping for self-driving vehicles: A survey,” *Machines*, vol. 12, no. 2, p. 118, 2024.
- [5] Y. Hao, J. Liu, Y. Liu, X. Liu, Z. Meng, and F. Xing, “Global visual-inertial localization for autonomous vehicles with pre-built map,” *Sensors*, vol. 23, no. 9, p. 4510, 2023.
- [6] T. Tuna, J. Nubert, Y. Nava, S. Khattak, and M. Hutter, “X-icp: Localizability-aware lidar registration for robust localization in extreme environments,” *IEEE Transactions on Robotics*, 2023.
- [7] D. Landry, F. Pomerleau, and P. Giguere, “Cello-3d: Estimating the covariance of icp in the real world,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8190–8196.
- [8] J. Nubert, E. Walther, S. Khattak, and M. Hutter, “Learning-based localizability estimation for robust lidar localization,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 17–24.
- [9] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.
- [10] K.-L. Low, “Linear least-squares optimization for point-to-plane icp surface registration,” *Chapel Hill, University of North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.
- [11] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, “Geometrically stable sampling for the icp algorithm,” in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*. IEEE, 2003, pp. 260–267.
- [12] T.-H. Kwok and K. Tang, “Improvements to the iterative closest point algorithm for shape registration in manufacturing,” *Journal of Manufacturing Science and Engineering*, vol. 138, no. 1, p. 011014, 2016.
- [13] Z. Rong and N. Michael, “Detection and prediction of near-term state estimation degradation via online nonlinear observability analysis,” in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2016, pp. 28–33.
- [14] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A.-a. Aghamohammadi, “Dare-slam: Degeneracy-aware and resilient loop closing in perceptually-degraded environments,” *Journal of Intelligent & Robotic Systems*, vol. 102, pp. 1–25, 2021.
- [15] S. Nobili, G. Tinchev, and M. Fallon, “Predicting alignment risk to prevent localization failure,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1003–1010.
- [16] D. Adolfsson, M. Magnusson, Q. Liao, A. J. Lilenthal, and H. Andreasson, “Coral—are the point clouds correctly aligned?” in *2021 European conference on mobile robots (ECMR)*. IEEE, 2021, pp. 1–7.
- [17] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, “Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization,” *Autonomous Robots*, pp. 1–21, 2022.
- [18] S. Zhao, H. Zhu, Y. Gao, B. Kim, Y. Qiu, A. M. Johnson, and S. Scherer, “Superloc: The key to robust lidar-inertial localization lies in predicting alignment risks,” *arXiv preprint arXiv:2412.02901*, 2024.
- [19] J. Xu, G. Huang, W. Yu, X. Zhang, L. Zhao, R. Li, S. Yuan, and L. Xie, “Selective kalman filter: When and how to fuse multi-sensor information to overcome degeneracy in slam,” *arXiv preprint arXiv:2412.17235*, 2024.
- [20] E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, “Factors to evaluate capability of map for vehicle localization,” *IEEE Access*, vol. 6, pp. 49 850–49 867, 2018.
- [21] Y. Endo, E. Javanmardi, and S. Kamijo, “Analysis of occlusion effects for map-based self-localization in urban areas,” *Sensors*, vol. 21, no. 15, p. 5196, 2021.
- [22] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, “Deep inference for covariance estimation: Learning gaussian noise models for state estimation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1436–1443.
- [23] Y. Fang, P. L. Yu, and Y. Tang, “Cnn-based realized covariance matrix forecasting,” *arXiv preprint arXiv:2107.10602*, 2021.
- [24] A. Censi, “An accurate closed-form estimate of icp’s covariance,” in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3167–3172.
- [25] W. Vega-Brown, A. Bachrach, A. Bry, J. Kelly, and N. Roy, “Cello: A fast algorithm for covariance estimation,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3160–3167.
- [26] A. De Maio and S. Lacroix, “Deep bayesian icp covariance estimation,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6519–6525.

- [27] J. Zhang, S. Singh *et al.*, “Loam: Lidar odometry and mapping in real-time.” in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [28] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, “Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation,” *arXiv preprint arXiv:2008.01550*, 2020.
- [29] C. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Neural Information Processing Systems*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1745976>
- [30] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.